

UNIVERSIDADE FEDERAL DO PARANÁ

CAROLINE MAZETTO MENDES

VISUALIZAÇÃO INTERATIVA DE MODELOS 3D REALISTAS NA WEB

CURITIBA

2014

CAROLINE MAZETTO MENDES

VISUALIZAÇÃO INTERATIVA DE MODELOS 3D REALISTAS NA WEB

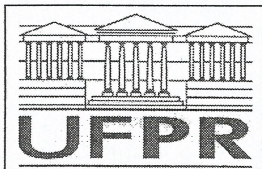
Tese apresentada como requisito parcial para obtenção do grau de Doutor. Programa de Pós-Graduação em Infomática, Setor de Ciências Exatas, Universidade Federal do Paraná.

Orientador: Prof. Dr. Luciano Silva

Coorientadora: Profa. Dra. Olga Regina Pereira Bellon

CURITIBA

2014



Ministério da Educação
Universidade Federal do Paraná
Programa de Pós-Graduação em Informática

PARECER

Nós, abaixo assinados, membros da Banca Examinadora da defesa da aluna de Doutorado em Ciência da Computação, Caroline Mazetto Mendes, avaliamos a tese de doutorado intitulada “*Visualização Interativa de Modelos 3D Realistas na Web*”, cuja defesa pública foi realizada no dia 16 de setembro de 2014, às 14:30 horas, no Departamento de Informática do Setor de Ciências Exatas da Universidade Federal do Paraná. Após avaliação, decidimos pela:

☒ aprovação da candidata. () reprovação da candidata.

Curitiba, 16 de setembro de 2014.

Prof. Dr. Luciano Silva
DINF/UFPR – Orientador

Profa. Dra. Olga Regina Pereira Bellon
DINF/UFPR – Coorientadora

Prof. Dr. Camilo Chang Dórea
UNB – Membro Externo

Prof. Dr. Joan Climent Vilaró
UPC/Espanha – Membro Externo

Prof. Dr. Sergio Scheer
UFPR – Membro Externo

Prof. Dr. Klaus de Geus
COPEL – Membro Externo

Profa. Dra. Marcia Cristina Rosato
MAE/UFPR – Membro Externo



AGRADECIMENTOS

Aos professores Luciano Silva e Olga R. P. Bellon pela orientação e amizade.

À CAPES pelo apoio financeiro para a realização deste trabalho.

A todos os colegas do IMAGO pelo incentivo e conhecimento compartilhado.

À Beatriz Trinchão Andrade de Carvalho pela amizade sincera.

Aos meus pais por terem investido com muito esforço na minha educação.

Ao Dyego Rogher Drees pelo apoio incondicional.

À Marília Mazetto Pesch pela inspiração de vida.

RESUMO

A visualização 3D remota de objetos culturais é uma ferramenta importante para diversas atividades na área de preservação digital, pois pode minimizar danos causados pelo transporte ou manipulação física dos objetos. Os modelos 3D gerados no processo de digitalização 3D podem auxiliar em restaurações ou na reconstrução física dos objetos. Devido aos avanços das tecnologias para digitalização 3D, os modelos 3D estão cada vez mais precisos. A disponibilização desses modelos para a visualização na web pode ser dificultada por restrições de acesso remoto, como por exemplo espaço de armazenamento, memória, capacidade de processamento e velocidade de conexão disponíveis. Nesse contexto, esta tese se concentra na visualização web interativa de modelos 3D realistas com fidelidade visual requerida ao estudo remoto dos objetos. O framework de visualização 3D baseada na web proposto incorpora a compressão progressiva de modelos 3D coloridos orientada por dados, uma base de dados de apoio à reconstrução 3D progressiva e uma aplicação web para a visualização 3D remota. Ao preservar as cores através do mapeamento de textura, a abordagem de compressão 3D progressiva orientada por dados proposta provê modelos intermediários com boa qualidade visual, principalmente a baixas resoluções. A estrutura do framework possibilita à aplicação suportar acessos simultâneos, além de viabilizar o acesso remoto por computadores e dispositivos móveis com restrição de recursos. A aplicação desenvolvida aproveita os recentes avanços das tecnologias web, proporcionando acesso livre de plugins e melhorando a qualidade da experiência do usuário. A solução proposta foi aplicada aos modelos 3D do conjunto de esculturas chamado “Os Doze Profetas” de Aleijadinho, considerado Patrimônio Mundial pela UNESCO. Através dos resultados experimentais pode-se comprovar que a preservação das cores combinada à reconstrução 3D progressiva é uma estratégia efetiva e apropriada para a visualização interativa de modelos 3D realistas na web.

Palavras-chave: Preservação digital. Modelos 3D realistas. Visualização na web. Reconstrução 3D progressiva. Qualidade visual.

ABSTRACT

Remote 3D visualization of valuable objects is a very important tool for the accomplishment of digital preservation tasks, since it prevents damages caused both by shipping and physical manipulation. The 3D models generated by 3D scanning may help physical restoration or reconstruction of the objects. Due to advances in 3D scanning technologies, the 3D models are becoming increasingly detailed. The web visualization of these models generated for digital preservation purposes is hampered by storage space, memory and processing restrictions. In this context, this thesis focuses on interactive web visualization of realistic 3D models for the remote study of the objects. The proposed web-based 3D visualization framework incorporates the data-driven progressive compression of colored 3D models, the database to support the progressive 3D reconstruction and a web application for remote 3D visualization. By preserving the colors through texture mapping, the proposed data-driven progressive 3D compression provides intermediate models with good visual quality at low resolutions. The framework enables the web application to support concurrent access, as well as providing access through computers with restricted resources. Our application takes advantages of the recent advances of the web technologies for providing a plugin-free experience. The proposed solution was applied to sculptures of the Twelve Prophets of Aleijadinho, considered World Heritage by UNESCO. The experimental results shows that the color preservation combined with progressive 3D reconstruction is an effective and appropriate strategy for interactive visualization of realistic 3D models on the web.

Keywords: Digital preservation. Realistic 3D models. Web-based visualization. Progressive 3D reconstruction. Visual Quality.

LISTA DE FIGURAS

FIGURA 1	– Visualização do fóssil de um <i>protocyon</i> (MENDES; SILVA; BELLON, 2012): (a) modelo 3D de baixa resolução renderizado localmente no cliente e (b) imagem do modelo 3D de alta resolução renderizado no servidor é sobreposta ao modelo de interação.	20
FIGURA 2	– Na compressão 3D progressiva, o modelo 3D de entrada é codificado iterativamente até obter um modelo base. Os dados da codificação obtidos em cada iteração compõem os dados de refinamento. A partir do modelo base, o modelo 3D é decodificado iterativamente utilizando os dados de refinamento. Nesse processo, são gerados modelos intermediários os quais são visualizados na reconstrução 3D progressiva.	25
FIGURA 3	– A operação <i>edge-collapse</i> substitui dois vértices adjacentes por um único vértice, enquanto a operação reversa <i>vertex-split</i> insere um vértice e duas novas faces. Fonte: (CHENG; OOI, 2008).	30
FIGURA 4	– Malha 3D original com vértices em vermelho selecionados para remoção em (a). Malha 3D após a remoção dos vértices e ret triangulação: (b) <i>patches</i> coloridos utilizando 3 cores em <i>4-coloring</i> e (c) <i>patches</i> coloridos utilizando 2 cores em <i>2-coloring</i> . Fonte: (COHEN-OR; LEVIN; REMEZ, 1999).	30
FIGURA 5	– Interface usada no experimento: imagem de referência posicionada ao topo e imagens de teste posicionadas abaixo. Fonte: (RAMANARAYANAN <i>et al.</i> , 2007).	39
FIGURA 6	– Um <i>patch</i> consiste em um vértice (amarelo) e suas faces incidentes, possui um <i>gate</i> de entrada (vermelho) e <i>gates</i> de saída (cinza escuro) que levam aos <i>patches</i> adjacentes. <i>Gates</i> são arestas com direção, as quais são representadas por dois vértices (direita e esquerda) e uma face frontal. Esquerda: <i>patch</i> com valência 5. Direita: <i>patch</i> nulo.	46
FIGURA 7	– Esquerda: vértice a ser removido em preto dentro do <i>patch</i> . Direita: cor marrom obtida na interpolação das cores dos vértices da face do meio. Fonte: (LEE; LAVOUÉ; DUPONT, 2012).	47
FIGURA 8	– Resultado visual da métrica proposta por Lee, Lavoué e Dupont (2012): (a) modelo 3D monkey e (b) vértices em vermelho impedidos de serem removidos durante a simplificação.	47
FIGURA 9	– Avaliação da qualidade visual do modelo 3D Joel para as abordagens de compressão. Curvas de distorção visual obtidas através do RMSE de Roy em (a) e do RMSE para uma vista em (b).	54
FIGURA 10	– Avaliação da qualidade visual do modelo 3D Anta para as abordagens de compressão. Curvas de distorção visual obtidas através do RMSE de Roy em (a) e do RMSE para uma vista em (b).	55
FIGURA 11	– Avaliação da qualidade visual do modelo 3D Índias para as abordagens de compressão. Curvas de distorção visual obtidas através do RMSE de Lindstrom e Turk em (a) e do RMSE de Roy em (b). ..	56

FIGURA 12	– Avaliação da qualidade visual do modelo 3D Baruc para as abordagens de compressão. Curvas de distorção visual obtidas através do RMSE de Lindstrom e Turk em (a) e do RMSE de Roy em (b). . .	58
FIGURA 13	– Fluxograma da abordagem de compressão 3D progressiva orientada por dados proposta. A partir do modelo 3D de entrada são geradas a textura colorida e a textura de células. Esta textura é usada para guiar o processo de simplificação e assim garantir o correto mapeamento da textura colorida para todos os modelos intermediários. A compressão progressiva tem como resultado um modelo base, dados de refinamento, uma textura de células e uma textura colorida.	64
FIGURA 14	– Visualização do modelo 3D Monkey: artefatos são renderizados se a textura colorida for aplicada aos modelos intermediários sem a simplificação orientada por dados de mapeamento de textura.	66
FIGURA 15	– Esquerda: <i>patch</i> com valência 6. Direita: <i>bounding box</i> englobando o <i>patch</i> com a nova configuração.	67
FIGURA 16	– Visualização do modelo 3D Monkey: modelos intermediários compartilhando a mesma textura colorida.	69
FIGURA 17	– Visualização do modelo 3D Anta: modelos intermediários compartilhando a mesma textura colorida.	69
FIGURA 18	– Visualização do modelo 3D Joel: modelos intermediários compartilhando a mesma textura colorida.	70
FIGURA 19	– Diferenças visuais entre os modelos base Monkey: (a) LLD, (b) LLD + métrica cor e (c) AD + CCC.	72
FIGURA 20	– Latência de visualização na reconstrução progressiva do modelo 3D Joel (vermelho) usando a textura colorida comprimida em JPEG progressivo (azul) e JPEG sequencial (rosa).	77
FIGURA 21	– Renderização do modelo 3D Joel com a textura comprimida com JPEG progressivo: (a) primeiro <i>scan</i> e (b) segundo <i>scan</i>	78
FIGURA 22	– Diagrama do modelo de dados usado para apoiar a reconstrução 3D progressiva.	82
FIGURA 23	– Diagrama da aplicação <i>web</i>	83
FIGURA 24	– Visualização de modelos 3D utilizando a aplicação <i>web</i> desenvolvida: (a) profeta Amós a partir de um computador e (b) profeta Baruc a partir de um <i>tablet</i>	84
FIGURA 25	– Visualização remota dos modelos 3D: (a) Joel e (b) Habacuc. Reconstrução 3D progressiva em vermelho e latência em azul. Latência considerando a transmissão do modelo 3D original comprimido com ZIP em rosa.	86
FIGURA 26	– Visualização de modelos intermediários na reconstrução progressiva do modelo 3D Habacuc.	87
FIGURA 27	– Visualização de modelos intermediários na reconstrução progressiva do modelo 3D Baruc.	87
FIGURA 28	– Tempo de resposta médio de 12 requisições na reconstrução progressiva do modelo 3D Habacuc. Comparação entre 50 e 100 acessos simultâneos ao servidor.	89
FIGURA 29	– Tempo médio da reconstrução total do modelo 3D Habacuc simulando: (a) 50 e (b) 100 acessos simultâneos.	89

FIGURA 30 – Tempo médio da reconstrução do modelo 3D Habacuc limitada em 20% de elementos simulando: (a) 50 e (b) 100 acessos simultâneos.	90
FIGURA 31 – Realização do experimento para a avaliação da qualidade visual subjetiva da reconstrução progressiva do modelo 3D Habacuc: (a) aviso sobre o início da etapa e (b) perguntas sobre a qualidade visual do modelo 3D.	94
FIGURA 32 – Diagrama do modelo de dados usado para armazenar informações e dados obtidos no experimento subjetivo.	95

LISTA DE TABELAS

TABELA 1	– Trabalhos relacionados à visualização de modelos 3D de acervos culturais. Descrição, tecnologias utilizadas e localização na Internet.	23
TABELA 2	– Cenários dos experimentos subjetivos para a avaliação da qualidade visual de modelos 3D - Parte I.	43
TABELA 3	– Cenários dos experimentos subjetivos para a avaliação da qualidade visual de modelos 3D - Parte II.	44
TABELA 4	– Taxas de compressão dos modelos testados em <i>bits</i> por vértice. Quantização usando 10 <i>bits</i>	52
TABELA 5	– Diferenças visuais entre as abordagens de compressão para modelos base com número similar de vértices. Renderização do modelo 3D Anta com cores e renderização em <i>wireframe</i>	53
TABELA 6	– Avaliação da qualidade visual do modelo 3D Joel para as abordagens de compressão. Comparação visual dos modelos intermediários com similar RMSE com o modelo 3D original.	54
TABELA 7	– Avaliação da qualidade visual do modelo 3D Anta para as abordagens de compressão. Comparação visual dos modelos intermediários com similar RMSE com o modelo 3D original.	56
TABELA 8	– Avaliação da qualidade visual do modelo 3D Índias para as abordagens de compressão. Comparação visual dos modelos intermediários com similar RMSE.	57
TABELA 9	– Avaliação da qualidade visual do modelo 3D Baruc para as abordagens de compressão. Comparação visual dos modelos intermediários com similar RMSE.	58
TABELA 10	– Informações sobre os modelos 3D testados: (a) número de vértices e faces do modelo 3D original; e (b) texturas de células e texturas coloridas.	69
TABELA 11	– Taxas de compressão dos modelos testados em <i>bits</i> por vértice. Quantização dos vértices usando 10 <i>bits</i>	71
TABELA 12	– Avaliação da qualidade visual do modelo 3D Monkey para as abordagens de compressão. Curvas de distorção visual obtidas através do RMSE vista a vista. Comparação dos modelos intermediários com similar porcentagem de elementos.	73
TABELA 13	– Avaliação da qualidade visual do modelo 3D Anta para as abordagens de compressão. Curvas de distorção visual obtidas através do RMSE vista a vista. Comparação dos modelos intermediários com similar porcentagem de elementos.	74
TABELA 14	– Avaliação da qualidade visual do modelo 3D Joel para as abordagens de compressão. Curvas de distorção visual obtidas através do RMSE vista a vista. Comparação dos modelos intermediários com similar porcentagem de elementos.	75
TABELA 15	– Custo para representar as coordenadas de textura: solução tradicional vs AD + CCC.	76

TABELA 16 – Resultado da compressão 3D progressiva orientada por dados: (a) textura de células e textura colorida; e (b) taxas de compressão dos modelos 3D testados. Quantização dos vértices usando 10 <i>bits</i> .	85
TABELA 17 – Latência de visualização: transmissão do modelo 3D original no formato ASCII sem e com compressão (ZIP) e compressão 3D progressiva.	86
TABELA 18 – Resultado do experimento para avaliação subjetiva da qualidade visual da reconstrução progressiva do modelo 3D Habacuc. BQV é a sigla para “boa qualidade visual”.	97

SUMÁRIO

1	INTRODUÇÃO	13
2	CONCEITOS	18
2.1	PRESERVAÇÃO DIGITAL 3D	18
2.2	VISUALIZAÇÃO DE MODELOS 3D NA WEB	19
2.2.1	Renderização Remota	19
2.2.2	Disponibilização do Conteúdo 3D	21
2.2.3	Considerações sobre Segurança	21
2.2.4	Tecnologias 3D	22
2.3	RECONSTRUÇÃO 3D PROGRESSIVA	24
2.3.1	Compressão 3D	24
2.3.2	Qualidade Visual	26
3	ESTADO DA ARTE	28
3.1	TRABALHOS RELACIONADOS À PRESERVAÇÃO DIGITAL 3D	28
3.2	MÉTODOS PROGRESSIVOS	29
3.2.1	Compressão 3D Progressiva	29
3.2.2	Compressão 3D Progressiva com Atributos de Aparência	32
3.2.3	Mapeamento de Texturas	33
3.3	AValiação DA QUALIDADE VISUAL DE MODELOS 3D	34
3.3.1	Métricas	34
3.3.2	Experimentos Subjetivos	36
3.4	CONCLUSÃO	41
4	RECONSTRUÇÃO 3D PROGRESSIVA: A CONTRIBUIÇÃO DA PRESERVAÇÃO DAS CORES NA VISUALIZAÇÃO DE MODELOS 3D REALISTAS.	45
4.1	COMPRESSÃO PROGRESSIVA DE MODELOS 3D COLORIDOS	45
4.1.1	Codificação	48
4.1.2	Decodificação	48
4.2	MÉTRICAS PARA A AVALIAÇÃO DA QUALIDADE VISUAL DE MODELOS 3D COLORIDOS	50
4.3	METODOLOGIA	50
4.4	RESULTADOS EXPERIMENTAIS	51
4.5	DISCUSSÃO DOS RESULTADOS E CONCLUSÃO	59
5	COMPRESSÃO 3D PROGRESSIVA ORIENTADA POR DADOS	62
5.1	VISÃO GERAL	63
5.2	CONSTRUÇÃO DAS TEXTURAS	64
5.3	COMPRESSÃO 3D PROGRESSIVA	65
5.3.1	Simplificação Orientada por Dados de Mapeamento de Textura	66
5.4	VISUALIZAÇÃO DOS MODELOS INTERMEDIÁRIOS	68
5.5	EXPERIMENTOS E COMPARAÇÕES	68
5.5.1	Taxa de Compressão	70
5.5.2	Qualidade Visual	71
5.5.3	Custo de Armazenamento	76
5.5.4	Carregamento Progressivo da Textura Colorida	76
5.6	CONCLUSÃO	78

6	FRAMEWORK DE VISUALIZAÇÃO 3D BASEADA NA WEB: UM CASO DE ESTUDO NA VISUALIZAÇÃO INTERATIVA DE MODELOS 3D DE ESCULTURAS DE ALEIJADINHO	80
6.1	DIGITALIZAÇÃO 3D	81
6.2	COMPRESSÃO 3D PROGRESSIVA ORIENTADA POR DADOS	81
6.3	BASE DE DADOS	81
6.4	APLICAÇÃO WEB	83
6.4.1	Reconstrução 3D Progressiva	83
6.5	RESULTADOS EXPERIMENTAIS	85
6.6	AVALIAÇÃO SUBJETIVA DA QUALIDADE VISUAL PARA A SELEÇÃO DOS NÍVEIS DE RECONSTRUÇÃO	91
6.6.1	Cenário	91
6.6.2	Procedimento	92
6.6.3	Base de Dados	93
6.6.4	Seleção das Etapas	95
6.6.5	Execução	95
6.6.6	Resultados	96
6.7	COMPARAÇÃO COM TRABALHOS SIMILARES	97
6.8	CONCLUSÃO	99
7	CONCLUSÕES E TRABALHOS FUTUROS	101
	REFERÊNCIAS	103

1 INTRODUÇÃO

A evolução dos dispositivos de aquisição e das tecnologias para reconstrução tridimensional (3D) de objetos tem sido expressiva na última década. Os modelos 3D gerados com a finalidade de preservação digital fornecem um registro do estado atual dos objetos, o qual poderá ser usado para comparações. Esses modelos podem auxiliar atividades de preservação digital, como restaurações ou a reconstrução física dos objetos em caso de desastres ou devido à própria degradação natural.

A visualização 3D remota permite investigar os objetos virtualmente com alto grau de realismo, evitando assim o transporte e a manipulação física que podem causar danos irreparáveis aos objetos. As exposições virtuais possibilitam uma nova forma de interação, na qual os usuários podem explorar detalhes dos objetos em variados ângulos, o que certamente não é permitido aos objetos físicos.

A disponibilização do conhecimento gerado à sociedade é uma importante etapa no processo de preservação digital. No entanto, existem poucos modelos 3D de objetos culturais em domínio público, onde possam ser acessados por estudantes ou pesquisadores (CIGNONI; SCOPIGNO, 2008). A preservação dos direitos autorais e a prevenção da distribuição não autorizada são fatores que contribuem para a falta desses modelos na Internet.

Os usuários têm expressado uma crescente necessidade em visualizar os modelos 3D na *web* e por meio de dispositivos móveis, como *tablets* e *smartphones*. A visualização *web* fornece a mesma experiência de uma aplicação *stand-alone*, porém por meio mais conveniente. Ela minimiza instalações no computador e permite a atualização automática da aplicação sem a intervenção do usuário. No entanto, os modelos 3D realistas não são um tipo de dados fácil de ser manipulado na *web*.

Esses modelos, geralmente representados por milhões de vértices e faces, demandam alto tempo de transmissão, espaço de armazenamento e poder de processamento. Se não há largura de banda suficiente, a aplicação *web* ficará congelada enquanto há dados a receber. A exibição de gráficos com aceleração 3D é uma recente capacidade dos navegadores *web* que ainda está sendo padronizada (EVANS *et al.*, 2014). Tecnologias *web*, como HTML e JavaScript, têm limitado poder computacional e impõem restrições no gerenciamento de memória.

A simplificação reduz a resolução do modelo 3D à custa da perda da precisão geométrica, porém viabiliza a sua disponibilização para visualização na *web*. Os esforços são concentrados na obtenção de uma versão simplificada tentando preservar a topologia do objeto. A versão que melhor compensa qualidade visual e alto nível de simplificação pode ser disponibilizada. Contudo, a qualidade visual de versões simplificadas pode não atender às expectativas dos usuários.

A renderização remota possibilita a visualização de grandes modelos 3D em computadores com recursos limitados (KOLLER *et al.*, 2004; MENDES; SILVA; BELLON, 2012; CRABLE, 2012). A estratégia consiste em disponibilizar uma versão altamente simplificada do modelo 3D para o usuário realizar a interação, enquanto imagens do mesmo modelo em alta resolução renderizado no servidor são sobrepostas ao modelo 3D de interação quando o usuário interrompe a movimentação. A boa experiência de visualização depende do tempo de resposta da aplicação (tempo compreendido entre logo após o envio da requisição até o recebimento de todos *bytes* da resposta, neste caso, uma imagem).

A disponibilização de modelos 3D progressivos reduz o tempo de espera do usuário para iniciar a visualização. Na compressão 3D progressiva, um modelo 3D de entrada é simplificado iterativamente até obter um modelo base suficientemente pequeno para ser transmitido na Internet. Nesse processo, são gerados modelos intermediários que podem ser visualizados na reconstrução 3D progressiva. Além de reduzir o espaço de armazenamento, esta estratégia possibilita ao usuário interagir com versões aproximadas enquanto o modelo 3D está sendo reconstruído.

As cores dos objetos fornecem informações sobre a cultura e arte de povos antigos. Nesse contexto, a preservação da aparência (*e.g.* cores) é requisito essencial para a visualização de aproximações com qualidade. No entanto, a maioria dos métodos de compressão progressiva consideram modelos 3D constituídos apenas por geometria e conectividade. São poucas as soluções para modelos 3D coloridos (LEE; LAVOUÉ; DUPONT, 2010, 2012). As cores também podem ser representadas por meio de texturas coloridas, porém, pelo nosso melhor conhecimento, não há uma solução para o mapeamento de texturas para compressão 3D progressiva.

A precisão geométrica é certamente importante para a preservação digital, por exemplo para o cálculo do volume do objeto e para a detecção de fendas ou

rachaduras com confiabilidade. No entanto, a partir de uma determinada resolução, distorções geométricas não são mais percebidas durante a visualização 3D (PAN; CHENG; BASU, 2005). A preservação das cores pode contribuir para melhoria da qualidade visual dos modelos 3D a baixas resoluções, permitindo a substituição do modelo 3D original por versões geometricamente menos precisas porém visualmente similares.

Infelizmente, ainda não há consenso entre a comunidade científica sobre como os modelos 3D podem ser comparados objetivamente, mesmo considerando apenas a forma geométrica (BERJÓN; MORÁN; MANJUNATHA, 2013), o que dificulta a avaliação da qualidade visual desses modelos. Uma maneira de avaliar a qualidade é através de experimentos subjetivos envolvendo observadores humanos (LAVOUÉ; CORSINI, 2010; BULBUL *et al.*, 2011; BERJÓN; MORÁN; MANJUNATHA, 2013). A avaliação subjetiva possui vantagens sobre as métricas objetivas, porém a aplicação de testes não é tarefa fácil para todas as aplicações (BULBUL *et al.*, 2011).

O objetivo principal deste trabalho é apresentar uma solução efetiva para a visualização de modelos 3D realistas na *web*, provendo a fidelidade visual requerida ao estudo remoto dos objetos culturais. Para tal, duas estratégias são combinadas: (1) a preservação das cores para melhorar a qualidade visual de modelos 3D coloridos e (2) a reconstrução 3D progressiva para visualização *web* interativa.

Um *framework* de visualização 3D baseada na *web* é proposto nesta tese. Este *framework* incorpora a compressão progressiva de modelos 3D coloridos orientada por dados, uma base de dados de apoio à reconstrução 3D progressiva e uma aplicação *web* para a visualização 3D remota. A estrutura do *framework* possibilita à aplicação suportar acessos simultâneos, além de viabilizar o acesso remoto por computadores e dispositivos móveis com restrição de recursos. A aplicação desenvolvida aproveita os recentes avanços das tecnologias *web*, proporcionando acesso livre de instalações (*e.g. plugins*). As principais contribuições desta tese são:

- Os experimentos apresentados no Capítulo 4 contribuem para a melhor compreensão dos desafios relacionados à avaliação da qualidade visual dos modelos 3D coloridos. Pelo nosso melhor conhecimento, não há estudos anteriores que analisam as métricas de qualidade objetivas aplicadas a esses modelos.

- Uma nova abordagem de compressão 3D progressiva orientada por dados de mapeamento de textura. O mapeamento de texturas para compressão 3D progressiva contribui para a melhoria da qualidade visual dos modelos intermediários a baixas resoluções.
- O desenvolvimento de um *framework* de visualização 3D baseada na *web* para a disponibilização de modelos 3D realistas. A solução proposta possibilita o estudo remoto dos objetos culturais com fidelidade visual para fins de ensino e pesquisa, e apoio às atividades de preservação digital.
- Uma metodologia para avaliação da qualidade visual subjetiva da reconstrução 3D progressiva. O objetivo é selecionar os níveis de reconstrução que apresentam qualidade visual suficiente de acordo com a opinião de observadores humanos.

Esta tese está organizada como descrita abaixo:

Capítulo 2 - Conceitos. Este capítulo abrange temas e conceitos relacionados a esta tese. A preservação digital 3D e as estratégias para a disponibilização de modelos 3D realistas para visualização na *web* são apresentados. Por fim, é apresentada a definição de reconstrução 3D progressiva.

Capítulo 3 - Estado da Arte. Este capítulo apresenta uma revisão dos trabalhos na literatura relacionados a esta tese. São apresentados os principais trabalhos relacionados à preservação digital 3D. Os principais métodos progressivos são apresentados, entre esses, enfoque aos que apresentam soluções para modelos coloridos ou texturizados. Este capítulo também descreve as métricas existentes para a avaliação da qualidade visual, além dos experimentos conduzidos para a avaliação da qualidade visual subjetiva dos modelos 3D.

Capítulo 4 - Reconstrução 3D Progressiva: A Contribuição da Preservação das Cores na Visualização de Modelos 3D Realistas. Este capítulo apresenta um conjunto de experimentos realizados com objetivo de analisar a contribuição da preservação das cores na reconstrução 3D progressiva.

Capítulo 5 - Compressão 3D Progressiva Orientada por Dados. Este capítulo apresenta uma nova abordagem para a compressão progressiva de modelos 3D co-

loridos, visando a melhoria da qualidade visual dos modelos intermediários a baixas resoluções.

Capítulo 6 - Framework de Visualização 3D Baseada na Web: Um Caso de Estudo na Visualização Interativa de Modelos 3D de Esculturas de Aleijadinho. Neste capítulo, o *framework* proposto é aplicado na visualização *web* de modelos 3D de esculturas feitas por Antônio Francisco Lisboa, conhecido como “O Aleijadinho”.

Capítulo 7 - Conclusão. Neste capítulo são apresentadas as conclusões e trabalhos futuros, seguido das referências bibliográficas.

2 CONCEITOS

Este capítulo abrange temas relacionados à visualização remota de modelos 3D realistas, como a preservação digital 3D e as estratégias para a disponibilização desses modelos na *web*. Por fim, é descrita a reconstrução 3D progressiva.

2.1 PRESERVAÇÃO DIGITAL 3D

A preservação digital de acervos tem como um dos seus principais objetivos a geração e a disponibilização de representações digitais de objetos que têm importância cultural ou científica. A digitalização 3D fornece um registro do estado atual dos objetos, o qual pode ser utilizado como referência para futuras comparações. Os modelos 3D gerados podem auxiliar restaurações e na reconstrução física dos objetos em casos de desastres ou devido a própria degradação natural.

Através da visualização 3D remota, os objetos podem ser explorados virtualmente com alto realismo, reduzindo a necessidade do transporte e da manipulação física que podem causar danos irreparáveis. A visualização 3D pode apoiar as atividades de preservação digital, como por exemplo a catalogação de acervos, análise da degradação do objeto, restaurações virtuais, entre outras (CIGNONI; SCOPIGNO, 2008).

Ao menos 3 etapas são necessárias para a digitalização 3D: (1) aquisição dos dados, (2) reconstrução 3D e (3) preservação da aparência. A superfície geométrica e as informações de aparência visual (*e.g.* cores e propriedades do objeto) são capturadas através de sistemas de aquisição compostos por dispositivos como *scanners* e câmeras fotográficas. A reconstrução 3D inicia com o alinhamento das vistas (obtidas a partir das imagens de profundidade) em um sistema de coordenadas comum, processo conhecido como alinhamento ou registro. Uma vez alinhadas, as vistas são combinadas em um único modelo 3D no processo de integração volumétrica. Por fim, texturas de alta resolução obtidas a partir de imagens coloridas são aplicadas ao modelo 3D.

Apesar do crescente avanço na geração de modelos 3D, poucos são acessíveis através da Internet. Uma vez transmitidos, os modelos ficam vulneráveis à pirataria e à violação dos direitos autorais. Os objetos podem ser reconstruídos fi-

sicamente a partir dos modelos 3D ou até mesmo das imagens exibidas durante a visualização. Por esse motivo, curadores de objetos valiosos e projetos de preservação digital podem estar relutantes em disponibilizar os modelos na Internet. Marca d'água 3D, criptografia e *pipeline* gráfico confiável são exemplos de técnicas de segurança (KOLLER; FRISCHER; HUMPHREYS, 2010). Essas técnicas dificultam roubos e permitem detectar a pirataria, porém ainda são necessárias soluções mais efetivas para impedir a reconstrução física dos objetos.

2.2 VISUALIZAÇÃO DE MODELOS 3D NA WEB

A visualização interativa de modelos 3D realistas na *web* pode ser dificultada por restrições de acesso remoto, como por exemplo espaço de armazenamento, memória, capacidade de processamento e velocidade de conexão disponíveis. Grandes modelos 3D (com milhões de vértices e faces) podem ser visualizados interativamente em ambientes controlados, como um computador local com todos os recursos previamente alocados (memória, processadores e placas de vídeo de última geração). A Internet no entanto possui natureza heterogênea, na qual estão conectados computadores com diferentes configurações de *hardware*, sistemas operacionais e navegadores *web*. A limitação de recursos computacionais e largura de banda insuficiente impactam negativamente na exibição interativa dos modelos 3D.

Apesar do crescente desenvolvimento da Internet, aplicações *web* podem não apresentar o mesmo desempenho de uma aplicação *stand-alone*. Em linguagens interpretadas como JavaScript, o gerenciamento de memória realizado pelo *garbage collector* pode causar sobrecarga de processamento e memória. Além disso, a exibição de gráficos 3D no navegador *web* ainda não foi totalmente padronizada. Independentemente da linguagem de programação ou da tecnologia 3D utilizada, uma aplicação *web* 3D interativa requer maior habilidade do desenvolvedor para obter bom desempenho. As principais estratégias para a visualização de modelos 3D realistas na *web* são apresentadas a seguir.

2.2.1 Renderização Remota

Apoiada por um sistema cliente-servidor, a renderização remota consiste em dedicar um servidor robusto para renderizar modelos 3D de alta resolução. Esses

modelos ficam protegidos no servidor e não são entregues ao cliente para visualização. Imagens dos modelos 3D renderizados no servidor são transmitidas para a visualização no cliente de acordo com a demanda. Entre as principais características dessa estratégia destacam-se o uso mínimo dos recursos computacionais do cliente, a segurança dos modelos 3D e fácil implementação para *web*. Essas características viabilizam a disponibilização de modelos 3D realistas quando há restrições de acesso remoto. No entanto, o tempo de resposta depende da latência da rede, da velocidade da conexão e da carga do servidor. A interrupção da conexão impossibilita a visualização dos detalhes do objeto a partir das imagens.

- **Visualização de imagens:** O usuário interage através de uma interface gráfica e visualiza imagens do modelo 3D pré-renderizado e/ou renderizado no servidor em tempo real (MENDES; SILVA; BELLON, 2012). Esta estratégia pode consumir muita largura de banda dependendo da frequência das interações.
- **Interação 3D + visualização de imagens:** O usuário interage com um modelo 3D de baixa resolução e visualiza imagens renderizadas no servidor em tempo real. Quando a movimentação é interrompida (*e.g.* ao soltar o botão do mouse), uma imagem do mesmo modelo 3D de alta resolução renderizado é sobreposta ao modelo de interação (KOLLER *et al.*, 2004; MENDES; SILVA; BELLON, 2012; CRABLE, 2012) (Figura 1).

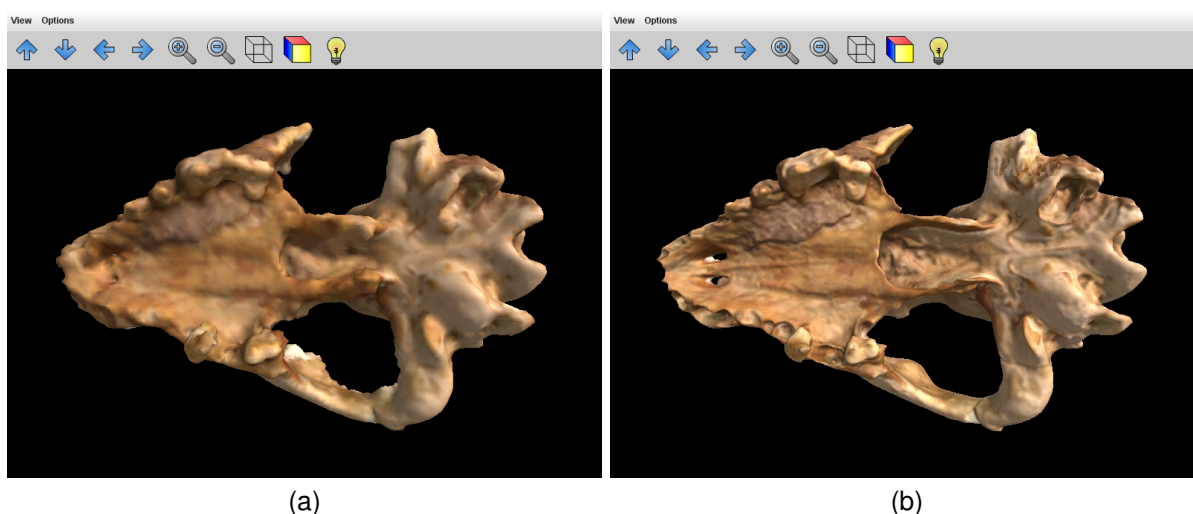


Figura 1 – Visualização do fóssil de um *protocyon* (MENDES; SILVA; BELLON, 2012): (a) modelo 3D de baixa resolução renderizado localmente no cliente e (b) imagem do modelo 3D de alta resolução renderizado no servidor é sobreposta ao modelo de interação.

2.2.2 Disponibilização do Conteúdo 3D

Diferente da renderização remota, esta estratégia possibilita a interação com modelos 3D com maior resolução. A disponibilização do conteúdo 3D pode ser total (modelo 3D original) ou parcial (versões simplificadas do modelo 3D original). A disponibilização parcial de modelos 3D realistas pode ser mais apropriada quando há restrições de acesso remoto.

- **Modelo 3D simplificado:** Um processo de simplificação é aplicado a fim de reduzir a complexidade e obter uma aproximação similar mantendo as características básicas do modelo 3D original. A simplificação provê uma única representação com resolução fixa, a qual pode não ser apropriada para todas as aplicações. Essa restrição motiva a próxima estratégia.
- **Modelo 3D Progressivo:** Consiste em disponibilizar uma versão altamente simplificada a partir da qual, operações de refinamento são aplicadas iterativamente até a reconstrução do modelo 3D. A reconstrução pode ser total ou parcial. O usuário pode interagir e visualizar versões aproximadas enquanto o modelo 3D é reconstruído progressivamente.

2.2.3 Considerações sobre Segurança

A renderização remota promove a segurança efetiva dos modelos 3D, uma vez que os modelos de alta resolução não são enviados para o computador do usuário. Técnicas de segurança podem ser incorporadas ao sistema de renderização remota, como por exemplo renderizar o modelo com pequenas modificações nos ângulos de visualização, inserir ruído nas imagens ou limitar a visualização para determinadas regiões do modelo 3D (KOLLER *et al.*, 2004).

Os arquivos acessados na *web* são armazenados em um diretório de arquivos temporários do navegador *web*. Por isso, não é recomendado disponibilizar o arquivo do modelo 3D original. A reconstrução 3D progressiva requer dados específicos com os quais é realizado o refinamento do modelo 3D. Esses dados sozinhos não caracterizam um modelo 3D, pois precisam estar associados ao nível de reconstrução correspondente. Portanto, esta estratégia não provê o acesso direto ao arquivo do modelo 3D original.

2.2.4 Tecnologias 3D

A VRML (Virtual Reality Modeling Language) possibilitou a criação das primeiras aplicações para visualização 3D na *web*. Ela permite descrever objetos e cenários 3D em um arquivo texto. Posteriormente foi sucedida pelo X3D, formato baseado no XML (Extensible Markup Language). Arquivos VRML/X3D podem ser visualizados no navegador *web* por meio de um *plugin*, como por exemplo o Octaga Player¹, Cosmo Player² e Cortona3D³. Apesar do formato ser bem difundido, *plugins* VRML/X3D não apresentam bom desempenho para renderizar modelos 3D realistas em tempo real (MENDES; SILVA; BELLON, 2012).

Outras tecnologias também baseadas em *plugins* como Flash e Applets Java permitem a criação de aplicações 3D para *web*. No entanto, a instalação de diferentes *plugins* para cada aplicação causa insatisfação nos usuários, os quais enfrentam problemas de incompatibilidades entre plataformas e navegadores *web*.

A WebGL (Web Graphics Library) é uma biblioteca JavaScript baseada no OpenGL ES 2.0 com a qual é possível exibir gráficos com aceleração 3D nativamente nos navegadores *web*, portanto não requer a instalação de um *plugin*. O Khronos Group entregou a primeira versão da especificação em março de 2011⁴. A WebGL é suportada pelas versões mais recentes dos navegadores *web* como Firefox, Opera e Google Chrome. Esse novo padrão para exibição de gráficos 3D na *web* possibilita a criação de aplicações verdadeiramente multiplataformas.

A Tabela 1 mostra os principais trabalhos relacionados à visualização de modelos 3D de acervos culturais. A maioria recorre a *plugins* VRML/X3D, porém trabalhos recentes estão investindo na criação de aplicações baseadas em WebGL. No entanto, como é possível observar, nem todos os trabalhos estão acessíveis para a visualização na *web*.

¹www.octaga.com

²ovrt.nist.gov/cosmo

³www.cortona3d.com

⁴www.khronos.org/news/press/khronos-releases-final-webgl-1.0-specification

Tabela 1 – Trabalhos relacionados à visualização de modelos 3D de acervos culturais. Descrição, tecnologias utilizadas e localização na Internet.

Descrição	Tecnologia 3D	Localização
Caso de estudo na digitalização e visualização <i>web</i> de objetos do Museu do Cairo, Egito (FAROUK <i>et al.</i> , 2003)	Flash Player e Quicktime	Eternal Egypt (1)
Modelagem e apresentação de construções e objetos da cidade de Praga, República Checa (JIRI; PAVEL, 2003)	VRML e Applets Java	The Virtual Old Prague (2)
Ferramenta para visualização <i>web</i> de sítios arqueológicos (GRABNER <i>et al.</i> , 2003)	ActiveX (IE)	-
Visualização interativa da Piazza dei Miracoli, Itália (CARROZZINO <i>et al.</i> , 2005)	XVR (<i>plugin</i> ActiveX)	Piazza dei Miracoli (3)
Reconstrução da antiga cidade de Xanthi, Grécia (KOUTSOUDIS; ARNAOUTOGLU; CHAMZAS, 2007)	VRML	The Old City of Xanthi (4)
Reconstrução e visualização do centro histórico da cidade de São Paulo, Brasil (CABRAL <i>et al.</i> , 2007)	X3D	-
Caso de estudo no desenvolvimento de uma aplicação <i>web</i> para consulta e exploração interativa de objetos culturais (GUARNIERI; PIROTTI; VETTORE, 2010)	X3D	-
Caso de estudo na criação do museu Virtual de Arte Ibérica (ROBLES-ORTEGA <i>et al.</i> , 2012)	X3D	-
Sistema de visualização do museu virtual 3D da Universidade Federal do Paraná (UFPR), Brasil (MENDES; SILVA; BELLON, 2012)	Applets Java	IMAGO (5)
Pipeline para produção de conteúdo 3D multimídia de objetos culturais (LARUE <i>et al.</i> , 2012)	WebGL	-
Sistema Venus 3D para publicação segura de grandes modelos 3D (CRABLE, 2012)	WebGL	CCRM Labs (6)
<i>Framework</i> para a transmissão de modelos de objetos culturais com funções de texturas bidirecionais (SCHWARTZ <i>et al.</i> , 2013)	WebGL	WebGL BTF Viewer Demo (7)
Projeto Aleijadinho 3D da Universidade de São Paulo (SCHWARTZ <i>et al.</i> , 2013)	Unity Web Player	Aleijadinho 3D (8)

(1) www.eternalegypt.org

(2) dcgi.felk.cvut.cz/cgg/vsp2

(3) piazza.opapisa.it/3D/index.html

(4) www.67100.gr/eng.htm

(5) www.imago.ufpr.br

(6) www.ccrmlabs.com/website2013/websitedemo_3.html

(7) btf.cs.uni-bonn.de/viewer

(8) www.aleijadinho3d.icmc.usp.br

2.3 RECONSTRUÇÃO 3D PROGRESSIVA

A reconstrução 3D progressiva consiste em um processo iterativo de refinamento do modelo 3D. A partir de uma versão simplificada, operações de refinamento são aplicadas à malha até a reconstrução total do modelo 3D. Esse processo requer um método para a geração de modelos 3D progressivos. Com apoio de um sistema cliente-servidor é possível limitar ou restringir os níveis de reconstrução devido a restrições de acesso remoto ou por motivos de segurança. Por exemplo, para a realização de pesquisas e estudo remoto dos objetos, a reconstrução progressiva pode alcançar níveis maiores que para realização de exposições virtuais públicas.

A Figura 2 ilustra a reconstrução 3D progressiva baseada em compressão. Na compressão 3D progressiva, um modelo 3D de entrada é codificado iterativamente até obter uma versão altamente simplificada chamada modelo base. Os dados da codificação obtidos em cada iteração compõem os dados de refinamento. A partir do modelo base, o modelo 3D é decodificado iterativamente utilizando os dados de refinamento. Nesse processo de compressão são gerados modelos intermediários os quais podem ser visualizados na reconstrução 3D progressiva. Mais informações são apresentadas a seguir.

2.3.1 Compressão 3D

A compressão 3D realiza a codificação das informações que compõem o modelo 3D, como geometria, conectividade e atributos de aparência. A geometria consiste nas posições dos vértices no espaço 3D e a conectividade descreve como os vértices estão conectados. Os atributos de aparência são os vetores normais, cores e coordenadas de textura.

Os métodos de compressão podem ser classificados como progressivos ou não progressivos (*single-rate*). Na compressão *single-rate*, o objetivo é remover a redundância diminuindo consideravelmente o tamanho do arquivo. O modelo 3D comprimido é representado por um único *stream* contendo os dados da sua codificação. Na compressão progressiva, o modelo 3D é codificado iterativamente até obter uma versão altamente simplificada (modelo base). O *stream* progressivo contém o modelo base e os dados da codificação obtidos em cada iteração.

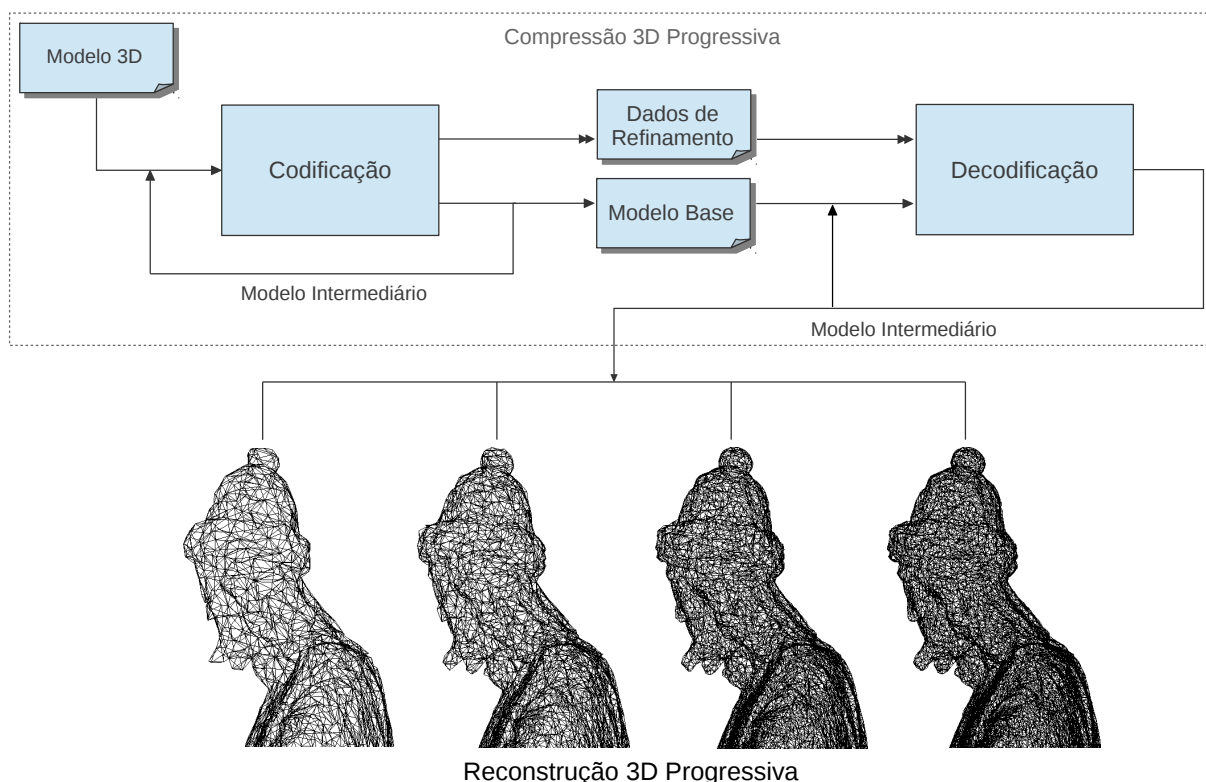


Figura 2 – Na compressão 3D progressiva, o modelo 3D de entrada é codificado iterativamente até obter um modelo base. Os dados da codificação obtidos em cada iteração compõem os dados de refinamento. A partir do modelo base, o modelo 3D é decodificado iterativamente utilizando os dados de refinamento. Nesse processo, são gerados modelos intermediários os quais são visualizados na reconstrução 3D progressiva.

Como descrito na Figura 2, a compressão progressiva oferece a possibilidade de visualizar aproximações (modelos intermediários) enquanto o modelo 3D é reconstruído progressivamente. Já na compressão *single-rate*, o modelo 3D pode ser visualizado apenas após a sua descompressão. Um dos objetivos da compressão progressiva é diminuir a latência de visualização, ou seja, diminuir o tempo em que o usuário espera para iniciar a visualização do modelo 3D. O desafio consiste em encontrar o equilíbrio entre a taxa de compressão e a qualidade visual da aproximação.

Os métodos de compressão podem ser orientados por conectividade ou por geometria de acordo com prioridade de codificação. Métodos de compressão progressiva orientados por geometria apresentam altas taxas de compressão, porém os modelos intermediários não apresentam boa qualidade visual, principalmente a baixas resoluções (MAGLO *et al.*, 2010). Nesse aspecto, os orientados por conectividade geram aproximações visualmente melhores.

A quantização⁵ pode ser usada para alcançar altas taxas de compressão. Os métodos de compressão progressiva orientados por conectividade têm apresentado técnicas para adaptar a quantização de cada modelo intermediário (LEE; LAVOUÉ; DUPONT, 2012). Os principais métodos de compressão 3D progressiva são apresentados no Capítulo 3.

2.3.2 Qualidade Visual

A imprecisão da geometria, a topologia deformada e a distorção das cores são aspectos que impactam negativamente a qualidade visual dos modelos intermediários. A maioria dos métodos de compressão progressiva consideram modelos 3D constituídos apenas por geometria e conectividade, e a importância visual das cores é negligenciada. Regiões delimitadas por descontinuidade de cor são visualmente significativas, portanto sua preservação é essencial para a visualização de aproximações com qualidade.

O mapeamento de texturas é uma maneira eficaz de preservar os detalhes visuais dos modelos 3D coloridos. No entanto, coordenadas de textura e mapa de textura para cada modelo intermediário requerem espaço de armazenamento significativo. Geralmente, as imagens de textura são comprimidas com o padrão JPEG, enquanto que as coordenadas de textura podem ser eficientemente codificadas com mecanismos de predição (ISENBURG; SNOEYINK, 2003; KIM *et al.*, 2005).

Para acessar a qualidade visual de um modelo intermediário, métricas baseadas em distância geométrica ou baseadas em imagens fornecem o erro de aproximação em relação ao modelo 3D original. Quanto menor o erro de aproximação, maior é a similaridade entre os modelos. Métricas baseadas em distância geométrica são as mais utilizadas para a avaliação de modelos 3D não coloridos.

Disponível no Metro Tool (CIGNONI; ROCCHINI; SCOPIGNO, 1998), a distância de Hausdorff mensura a similaridade entre dois conjuntos de pontos A e B :

$$H(A,B) = \max(D(A,B), D(B,A)) \text{ onde } D(A,B) = \max_{a \in A}(\min_{b \in B}(\|a - b\|)) \quad (1)$$

⁵Na compressão 3D, quantização consiste em reduzir o número de *bits* usado para representar a posição dos vértices.

A distância entre A e B é definida como a maior entre as menores distâncias euclidianas calculadas para cada ponto de A em relação a todos os pontos de B. Assim, a distância de Hausdorff computa a distância final entre dois conjuntos como o máximo entre as distâncias $D(A, B)$ e $D(B, A)$.

Outra abordagem consiste em comparar imagens 2D do modelo intermediário e do modelo 3D original renderizados sob mesmos pontos de vista. O Root Mean Square Error (RMSE) calcula a distância euclidiana *pixel a pixel* entre duas imagens X e Y :

$$RMSE(X, Y) = \sqrt{\frac{1}{mn} \sum_{i=1}^m \sum_{j=1}^n (x_{ij} - y_{ij})^2} \quad (2)$$

A comparação entre duas imagens pode não capturar toda a aparência do objeto. O ideal é obter um conjunto de vistas movimentando a câmera em torno do modelo 3D e combinar as imagens das vistas em uma única medida de erro. Ainda assim, esse conjunto de vistas pode não ser suficiente. As principais métricas e abordagens para a avaliação da qualidade visual de modelos 3D são descritas no Capítulo 3.

3 ESTADO DA ARTE

Neste capítulo são apresentados os principais trabalhos relacionados à preservação digital e seus esforços para a disponibilização de modelos 3D na Internet. Também são apresentados os métodos progressivos, entre esses, os que apresentam soluções para modelos 3D com atributos de aparência. Este capítulo também apresenta as métricas existentes para a avaliação da qualidade visual dos modelos 3D e experimentos realizados para a avaliação da qualidade visual subjetiva.

3.1 TRABALHOS RELACIONADOS À PRESERVAÇÃO DIGITAL 3D

No projeto Michelangelo Digital (LEVOY *et al.*, 2000) foram digitalizadas grandes obras de arte feitas por Michelangelo, com destaque para a estátua de Davi. O sistema de aquisição incorporou diferentes dispositivos, como *scanner* de triangulação *laser* e câmeras digitais. Devido à importância dos objetos, o projeto focou na disponibilização segura a fim de evitar a reconstrução física dos objetos. Koller *et al.* (2004) apresentaram a técnica de renderização remota, a qual permite ao usuário interagir com modelos 3D de baixa resolução e visualizar imagens do mesmo modelo 3D de alta resolução renderizado no servidor. Os modelos 3D podem ser visualizados utilizando o *software* ScanView (graphics.stanford.edu/software/scanview).

Farouk *et al.* (2003) desenvolveram um sistema para digitalizar objetos do Museu Egípcio localizado no Cairo. O projeto desenvolveu um *pipeline* para geração de modelos 3D usando dispositivos para a captura das imagens de profundidade, além de um sistema de fotometria. Os modelos 3D resultantes são usados para restauração virtual e podem ser visualizados no *website* Eternal Egypt (www.eternalegypt.org).

Koutsoudis, Arnaoutoglou e Chamzas (2007) realizaram a reconstrução 3D da antiga cidade de Xanthi, na Grécia, utilizando equipamentos de baixo custo. O objetivo foi a exibição do patrimônio cultural utilizando modelos 3D com texturas detalhadas. Primeiramente foi realizado um estudo exaustivo para identificar as características geométricas básicas da cidade. Nesse estudo foram capturados diversos pontos de vistas utilizando equipamentos de filmagem, como câmeras digitais, para depois obter as coordenadas 3D dos pontos usando fotometria. O modelo 3D final é disponibilizado para visualização no formato VRML (www.67100.gr/eng.htm). Os dados de geometria

e texturas são compactados para facilitar a transmissão. As texturas são carregadas progressivamente de acordo com a exibição da geometria, permitindo assim a interação mesmo enquanto elas estão sendo carregadas.

O Grupo IMAGO de Pesquisa¹ da Universidade Federal do Paraná (UFPR) tem realizado a preservação digital e a visualização 3D de objetos de diversas instituições, como o Museu de Arqueologia e Etnologia da UFPR (MAE), Museu Metropolitan de Arte de Curitiba, Museu de Ciências Naturais da UFPR, entre outras. Em parceria com o IPHAN (Instituto do Patrimônio Histórico e Artístico Nacional) e a UNESCO, o grupo realizou a digitalização 3D da obra Os Doze Profetas de Aleijadinho (ANDRADE *et al.*, 2011). O processo de preservação digital incorpora as etapas de aquisição utilizando *scanners* e câmeras fotográficas, reconstrução 3D e geração de texturas de alta qualidade. Os modelos 3D do acervo são disponibilizados para visualização na *web* no Museu Virtual 3D do IMAGO (MENDES; SILVA; BELLON, 2012).

3.2 MÉTODOS PROGRESSIVOS

3.2.1 Compressão 3D Progressiva

Métodos de compressão 3D progressiva oferecem a possibilidade de visualizar modelos intermediários durante a reconstrução do modelo 3D. Hoppe (1996) introduziu o primeiro método progressivo, chamado *Progressive Meshes*. O método consiste na aplicação de sucessivas operações na malha 3D as quais podem ser revertidas. O modelo 3D de entrada é simplificado aplicando-se uma sequência de operações *edge-collapse* até obter o modelo base. A partir deste, o modelo 3D é reconstruído progressivamente com a operação reversa *vertex-split* (Figura 3).

O *Progressive Meshes* possibilita a geração de diversos níveis de detalhes, porém apresenta baixa taxa de compressão. A operação *vertex-split* requer o armazenamento dos dados de adjacência, os índices de todos os vértices da operação e a posição do novo vértice. Variações foram propostas a fim de melhorar a taxa de compressão (TAUBIN *et al.*, 1998; COHEN-OR; LEVIN; REMEZ, 1999; PAJAROLA; ROSSIGNAC, 2000; ALLIEZ; DESBRUN, 2001).

¹www.imago.ufpr.br

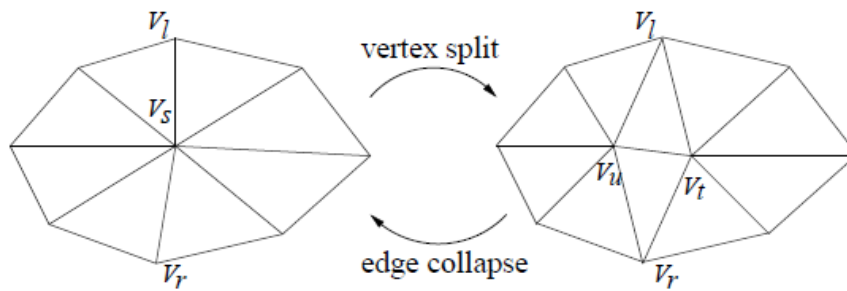


Figura 3 – A operação *edge-collapse* substitui dois vértices adjacentes por um único vértice, enquanto a operação reversa *vertex-split* insere um vértice e duas novas faces. Fonte: (CHENG; OOI, 2008).

Taubin *et al.* (1998) apresentaram uma representação de refinamento chamada *Progressive Forest Split*. Na operação *forest split*, a malha 3D é refinada realizando um corte através da floresta de arestas, seguido pela duplicação dos vértices da borda resultante. Assim, as árvores resultantes são preenchidas com polígonos simples e por fim, os novos vértices são deslocados para as novas posições.

Cohen-Or, Levin e Remez (1999) propuseram duas técnicas chamadas *4-coloring* e *2-coloring*. *Patches* (um vértice e suas faces incidentes) são dizimados pela operação de remoção do vértice central e depois são retriangulados. Os triângulos do *patch* são coloridos de forma a permitir sua localização. Na técnica *4-coloring*, *patches* são coloridos usando 3 cores diferentes, enquanto na técnica *2-coloring* são usadas apenas 2 cores desde que o *patch* tenha sido retriangulado no formato da letra Z (Figura 4). A posição do novo vértice inserido dentro do *patch* é predizível usando os vértices vizinhos.

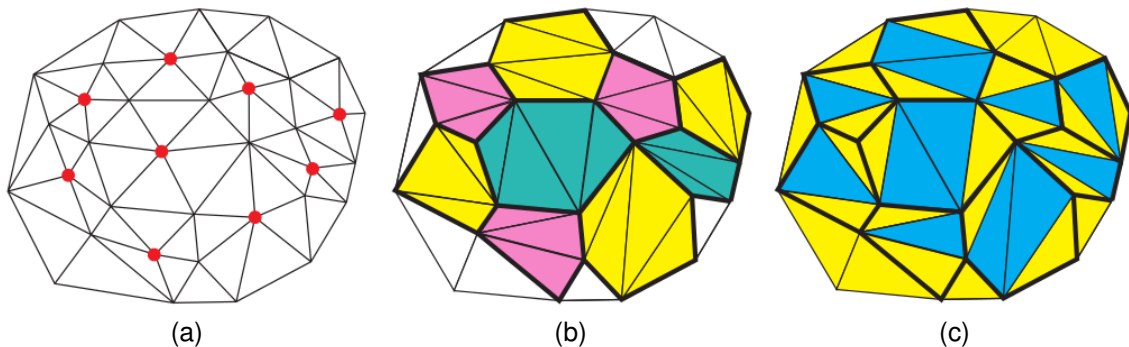


Figura 4 – Malha 3D original com vértices em vermelho selecionados para remoção em (a). Malha 3D após a remoção dos vértices e retriangulação: (b) *patches* coloridos utilizando 3 cores em *4-coloring* e (c) *patches* coloridos utilizando 2 cores em *2-coloring*. Fonte: (COHEN-OR; LEVIN; REMEZ, 1999).

Pajarola e Rossignac (2000) apresentaram uma variação chamada *Compressed Progressive Mesh*. Os *vertex-splits* são agrupados em conjuntos independentes (*batches*) e os vértices da operação são marcados ao custo de 1 *bit*. A posição do novo vértice é obtida por meio da combinação linear dos vértices vizinhos.

Após observar a compacta distribuição da valência dos vértices, Alliez e Desbrun (2001) propuseram um método de compressão orientado à valência com excelente taxa de compressão (média de 3,7 *bits* por vértice). Na etapa de codificação é feita a conquista de *patches* (um vértice e suas faces incidentes) para dizimação. *Patches* com valência entre 3 e 6 são dizimados. Na decodificação, é feito o mesmo percurso a fim de encontrar os *patches*, seguido pela adição de um novo vértice e duas novas faces em cada *patch*. A diferença da posição do vértice removido e do baricentro do *patch* é codificado em coordenadas Frenet Frame local².

Os métodos previamente apresentados são classificados como orientados à conectividade. Gandoïn e Devillers (2002) propuseram um método orientado à geometria baseado em subdivisão kd-tree. Posteriormente, Peng e Kuo (2005) apresentaram um melhoramento desse método substituindo a kd-tree por octree. Esses métodos apresentam bons resultados na compressão sem perdas. Contudo, como observado por Maglo *et al.* (2010), métodos orientados à geometria proveem modelos intermediários com pobre qualidade visual comparados aos orientados à conectividade. Abordagens mais recentes (VALETTE; CHAINE; PROST, 2009; LEE; LAVOUÉ; DUPONT, 2012) focam na adaptação da precisão da quantização da geometria para otimizar a taxa de distorção.

Valette, Chaine e Prost (2009) apresentaram uma abordagem chamada Incremental Parametric Refinement (IPR), que diferente das abordagens apresentadas previamente, a conectividade não é controlada em primeira etapa. A conectividade do modelo 3D original é garantida apenas após a reconstrução do nível de resolução mais alto. Essa abordagem também adapta a precisão da quantização para cada vértice baseado na configuração da geometria local. A abordagem produz resultados competitivos em termos de taxa de distorção comparado a trabalhos anteriores (*e.g.* (PENG; KUO, 2005)).

²Coordenadas derivadas dos vetores unitários tangente, normal e binormal (triedro de Frenet) calculados para o *patch*.

Lee, Lavoué e Dupont (2012) adaptam a precisão da quantização dos dados de geometria e cor para cada modelo intermediário. Essa abordagem é baseada no método de Alliez e Desbrun (2001) e em seu trabalho anterior para codificação progressiva de modelos 3D coloridos (LEE; LAVOUÉ; DUPONT, 2010). O método proposto previne a dizimação de vértices que possam eventualmente causar distorção visual. Os resultados mostram que o método melhora o desempenho na taxa de distorção para modelos 3D com atributo de cor e pode competir com métodos recentes como o IPR para modelos 3D sem atributos.

3.2.2 Compressão 3D Progressiva com Atributos de Aparência

Cores e texturas contribuem significativamente na aparência visual dos modelos 3D. No entanto, há poucos métodos de compressão com atributos de aparência associado. Cai, Qi e Shen (2007) propuseram um método de compressão baseado em octree que permite codificar as coordenadas de textura. As imagens de textura de cada modelo intermediário são transmitidas para o cliente antes de iniciar a decodificação. O trabalho não menciona o processo de mapeamento de textura utilizado.

Cirio, Lavoué e Dupont (2010) propuseram um *framework* que usa atributo de cor para guiar o processo de compressão 3D progressiva baseado em subdivisão em kd-tree. O processo de divisão celular é feito de acordo com o atributo de cor associado aos vértices da célula. O método proposto é mais apropriado à compressão de modelos 3D onde a geometria e conectividade são objetivos secundários, melhorando a qualidade visual comparada à sua versão não guiada pela cor.

Lee, Lavoué e Dupont (2010) apresentaram dois métodos para a codificação progressiva das cores baseado no método de Alliez e Desbrun (2001). No primeiro método, o vértice da borda do *patch* que possui cor mais próxima à cor do vértice removido é selecionado, e a diferença dessas cores é codificada no espaço CIE Lab. No segundo método, os vértices do modelo de entrada são agrupados de acordo com as cores. Assim, as cores dos vértices são substituídas pela cor representativa mais próxima usando uma tabela de mapeamento.

Recentemente, Lee, Lavoué e Dupont (2012) propuseram um método de compressão baseado na otimização da taxa de distorção para modelos sem e com atributo de cor. Essa abordagem é baseada no método de Alliez e Desbrun (2001) e em seu

trabalho anterior para codificação progressiva das cores (LEE; LAVOUÉ; DUPONT, 2010). A simplificação orientada por dados (atributo de cor) previne a dizimação de vértices que podem eventualmente causar distorção visual.

3.2.3 Mapeamento de Texturas

Sander *et al.* (2001) apresentaram um método no qual uma sequência de modelos *Progressive Meshes* compartilha uma mesma parametrização de textura. O modelo 3D de entrada é particionado em regiões planas, as quais são parametrizadas de forma à minimizar estiramentos e desvios de textura. Para garantir a conformidade das regiões, não é realizada a operação *edge-collapse* em vértices de borda durante a simplificação.

Chen e Chuang (2006) propuseram uma abordagem que adapta o mapeamento do conteúdo da textura a cada operação *edge-collapse*, porém um único mapa de textura é requerido para toda sequência de modelos 3D. As coordenadas de texturas são derivadas a partir de um mapa de indexação. A adaptação de textura é invertível, portanto também é feita a cada operação *vertex-split* durante o refinamento.

Ambas as soluções apresentadas acima possuem características que dificultam o seu uso na compressão 3D progressiva e na visualização *web*. A restrição à remoção dos vértices de bordas das regiões no método de Sander *et al.* (2001) penaliza a taxa de compressão. Além disso, vértices de borda precisam armazenar a sua coordenada de textura em cada região. Para manter as texturas de cada nível de detalhes em um único mapa de textura físico, o método de Chen e Chuang (2006) requer um mapa temporário para suportar as adaptações no mapa de textura e no mapa de indexação.

Soluções existentes para o mapeamento de textura de modelos 3D não progressivos podem ser adaptadas à compressão progressiva. Novas abordagens usam dados de geometria no mapeamento de texturas. Nesse caso, as coordenadas de textura são calculadas diretamente a partir da superfície da malha 3D. Texturas Oc-tree (BENSON; DAVIS, 2002; LEFEBVRE; HORNUS; NEYRET, 2005; LACOSTE *et al.*, 2007) mapeiam as cores da textura usando octrees a fim de reduzir o espaço de armazenamento. Texturas TileTrees (LEFEBVRE; DACHSBACHER, 2007) também usam octrees para o mapeamento de texturas. Conjuntos de cores são mapeados

dentro de folhas octrees e são compactamente armazenados em uma textura 2D. Em geral, os métodos baseados em octree são implementados em GPU³, ou seja, eles carregam a octree em uma textura a qual é acessada pela GPU. Contudo, os incrementos na profundidade da octree resulta em acessos adicionais à memória, o qual é custoso para renderização em GPU.

Apaza-Agüero, Silva e Bellon (2011) propuseram um método que usa complexos de células cúbicas (estrutura similar à octree). A estrutura de células cúbicas é armazenada em uma pequena textura, a partir da qual são geradas as coordenadas de textura durante a renderização em GPU. Uma das vantagens deste método é que a estrutura de células cúbicas pode ser acessada diretamente, diferente dos baseados em octree.

3.3 AVALIAÇÃO DA QUALIDADE VISUAL DE MODELOS 3D

3.3.1 Métricas

A maioria das métricas para avaliação da qualidade visual trabalham sobre a geometria do modelo 3D. Outra abordagem possível é utilizar imagens 2D do modelo 3D renderizado em determinados pontos de vista. Métricas objetivas podem considerar princípios do sistema visual humano, trabalhando sobre diferentes aspectos de percepção.

Duas métricas objetivas baseadas em distância geométrica bem conhecidas são o RMSE e a distância de Hausdorff (Equação 1). O RMSE calcula a distância euclidiana entre duas malhas 3D com mesma conectividade, enquanto a distância de Hausdorff mensura a similaridade entre dois conjuntos.

Métricas baseadas em rugosidade partem do princípio que ruídos presentes na malha 3D podem ser facilmente detectados em superfícies suaves. Wu *et al.* (2001) apresentaram uma métrica que calcula a rugosidade por meio do ângulo diedro formado entre faces adjacentes. Corsini *et al.* (2007) propuseram duas métricas baseadas em rugosidade. A primeira é uma variação de Wu *et al.* (2001), enquanto a segunda mensura a rugosidade de uma malha 3D como a diferença entre esta e sua versão suavizada. Essas métricas foram utilizadas para avaliar a distorção visual causada pela adição de marca d'água.

³Graphics Processing Unit. Unidade de processamento gráfico da placa de vídeo.

Recentemente, Wang, Torkhani e Montanvert (2012) apresentaram uma métrica baseada em rugosidade chamada FMDP (*Fast Mesh Perceptual Distance*). Para os dois modelos 3D a serem comparados, primeiro é feito o cálculo da rugosidade local para cada vértice baseado em curvatura gaussiana e depois é calculada a rugosidade global usando integrais de superfície normalizada. Por fim, a distância entre os dois modelos 3D é dada pela diferença entre as duas integrais de superfície.

Lee, Varshney e Jacobs (2005) utilizam o conceito de saliência da malha 3D. A ideia principal é a identificação de regiões mais importantes as quais visualmente atraem mais a atenção, como regiões que diferem fortemente da sua vizinhança. A saliência da malha 3D é extraída computando mapas de curvatura em múltiplas escalas. A métrica pode ser usada para decidir regiões a serem preservadas durante a simplificação 3D.

Lavoué *et al.* (2006) propuseram a métrica MSDM (Mesh Structural Distortion Measure) e posteriormente Lavoué (2011) apresentou a MSDM2. Ambas são baseadas em distorção estrutural, a qual considera a teoria que o sistema visual humano é sensível a certos padrões visuais e consegue extrair bem as informações estruturais de uma cena. As informações estruturais são capturadas usando mapa de distorção em diferentes escalas. A métrica proposta pode ser usada para o cálculo da taxa de distorção para métodos de compressão 3D progressiva.

Bian, Hu e Martin (2009) propuseram uma métrica baseada na teoria de campo de tensão (*Strain Field*). A métrica considera os modelos 3D como objetos com elasticidade e as deformações na malha 3D são quantificadas usando cálculo do campo de energia (*Strain Energy*). Os experimentos realizados mostraram que os observadores percebem a diferença entre os modelos 3D mais facilmente quando a distorção em um local é intensa. A métrica proposta pode ser usada para guiar aplicação de marca d'água 3D.

Roy, Fofou e Truchetet (2004) propuseram uma métrica baseada em atributos, como por exemplo normais, cores ou coordenadas de textura. A métrica calcula as diferenças locais entre os atributos de duas malhas. A técnica auxilia a simplificação de modelos 3D com mínima distorção visual ao considerar atributos de aparência.

Cheng e Boulanger (2005) propuseram uma métrica baseada em JND (*Just Noticeable Difference*). JND significa a mudança mínima em valor de percepção a

fim de produzir uma variação visual que pode ser notada. A métrica integrada com análise de percepção foi utilizada para avaliar a qualidade visual dos algoritmos de simplificação 3D baseados em LOD⁴ (*Progressive Meshes e Quadric Error Metric*).

Pan, Cheng e Basu (2005) realizaram um estudo sobre como a resolução da geometria e da textura contribuem para a qualidade visual do modelo 3D. A métrica proposta permite estimar a qualidade visual para otimizar a transmissão de modelos 3D texturizados em ambientes de rede restritos. Os resultados dos experimentos realizados sugerem que observadores são mais sensíveis a distorções da textura do que da geometria.

Tian e AlRegib (2004) propuseram uma métrica de qualidade chamada FQM (*Fast Quality Measure*), que combina os cálculos de distância geométrica e distorção de textura (*mean square texture deviation e mean square image error*). A métrica permite selecionar as resoluções da geometria e da textura a fim de maximizar a qualidade visual para transmissão em rede com largura de banda limitada.

A métrica proposta por Zhu *et al.* (2010) funciona como um indicador quantitativo da percepção visual. É dada pela relação entre o número de *pixels* representando os detalhes perceptíveis e o total de *pixels* na imagem do modelo 3D renderizado. A métrica é utilizada para geração de LODs de fachadas de edifícios de forma otimizada, minimizando as redundâncias entre os níveis.

3.3.2 Experimentos Subjetivos

Rogowitz e Rushmeier (2001) conduziram um experimento para avaliar se métricas baseadas em imagens são adequadas para capturar a qualidade perceptível de modelos 3D. Foram utilizados dois modelos 3D coloridos em cinza (Bunny e Happy Buddha) mais duas versões simplificadas de cada um. O modelo 3D original foi posicionado na tela ao lado das suas duas versões simplificadas sob duas condições de iluminação diferentes. Os observadores julgaram a qualidade perceptível comparando imagens estáticas do modelo 3D e uma sequência animada de imagens (rotação). Para cada teste, a classificação escolhida foi de 0 (pior) a 10 (perfeito). Observadores tinham a liberdade de visualizar todas as vistas ou examinar quadros individuais no caso das animações. Os resultados mostraram grande diferença na qualidade

⁴Level of Detail. Técnica para representar modelos 3D em níveis de detalhes.

perceptível dependendo da direção da luz. Foi observado também que o objeto em movimento (animação) reduz a detecção de artefatos produzidos pela simplificação. A conclusão foi que mesmo que uma métrica possa caracterizar a qualidade perceptível no modelo 3D utilizando imagens estáticas, ela pode não funcionar para uma sequência animada das imagens de projeção.

Watson, Friedman e McGaffey (2001) realizaram um estudo sobre as técnicas para mensurar a fidelidade visual de modelos 3D simplificados. As técnicas examinadas foram: tempo de nomeação, classificação e preferência (escolha do melhor estímulo). Foram utilizados 36 modelos 3D de objetos feitos por humanos e representação de animais, sem atributos como cores ou normais. Cada modelo 3D foi simplificado em dois níveis utilizando dois algoritmos de simplificação (Vclust e QSlm). Uma imagem do modelo 3D original foi posicionado no topo da tela, seguida abaixo pela imagem de duas versões simplificadas com mesmo nível, porém geradas com os dois algoritmos. No teste do tempo de nomeação, foi avaliado quão rápido os participantes respondem qual o nome de cada objeto. Na classificação os participantes atribuem números em uma escala de 7 pontos, enquanto no teste de preferências, eles escolhem qual imagem representa melhor o objeto. Os resultados mostraram que os modelos 3D de animais são nomeados mais lentamente que os de objetos feitos por humanos e que o tempo de nomeação aumenta quanto maior o nível de simplificação. Resultados de classificação e preferência indicaram que o QSlm geralmente é o melhor simplificador, principalmente para os modelos 3D de animais, enquanto o Vclust é mais efetivo para os objetos feitos por humanos.

Rushmeier, Rogowitz e Piatko (2000) realizaram um experimento para examinar a qualidade visual de modelos 3D com textura. Foram utilizados dois objetos: uma esfera com superfície suave e uma esfera com superfície rugosa. Para cada objeto foram gerados dois níveis de simplificação (versões média e simples). Para o estímulo foram utilizados: modelos 3D em versões completa, média e simples; texturas em versões completa (512x512), média (256x256) e simples (64x64); sob duas condições de iluminação diferentes. Na tela foram posicionadas doze imagens geradas das combinações de geometria e textura (incluindo geometria sem textura) para cada condição de iluminação, utilizando fundo preto. Oito observadores classificaram cada imagem usando uma escala de 0 (pior) a 100 (perfeito). Os resultados no geral indicaram que

há benefícios em substituir detalhes da geometria utilizando textura, desde que ela seja em alta resolução.

Pan, Cheng e Basu (2005) realizaram um experimento para prover uma métrica de qualidade visual. O experimento permitiu entender a importância da resolução da textura e da resolução da geometria na avaliação da qualidade global dos modelos 3D. Foram utilizados cinco objetos visualmente diferentes. Para cada modelo 3D foram gerados seis níveis de resoluções em *wireframe* e três níveis de resolução de textura. O modelo 3D de maior resolução (geometria e textura) foi posicionado à esquerda na tela, o de menor à direita e o modelo 3D de teste ao centro. Os participantes compararam a qualidade do modelo 3D de teste com os dois referenciais, classificando-o como muito pobre (1) a muito bom (5). Os modelos 3D de referência foram rotados à mesma velocidade do modelo 3D de teste, ao qual foi permitido ajustar a rotação.

Corsini *et al.* (2007) realizaram dois experimentos para validar as duas métricas de qualidade visual propostas para modelos 3D com marca d'água. O primeiro experimento visou incorporar dados subjetivos de percepção visual nas métricas objetivas, enquanto o segundo foi realizado para validá-las. Foram utilizados quatro modelos 3D, e para cada um foi gerada uma versão utilizando três algoritmos de marca d'água diferentes para três níveis de resolução. O modelo 3D original e com marca d'água foram apresentados na tela, e aos participantes foi permitido interagir usando rotação e *zoom*. A cor de fundo utilizada foi degradê de azul (cima) ao branco (baixo), a fim de não sobrestimar o contorno do modelo 3D em relação a sua forma geral. Com o objetivo de diminuir a influência da percepção humana sobre a cor do objeto, foi atribuída uma cor de material para obter efeito de "pedra" ao modelo 3D. Os participantes atribuíram valores de 0 a 10 para indicar a distorção (artefatos percebidos).

Ramanarayanan *et al.* (2007) conduziram uma série de experimentos para estudar como geometria, material e iluminação interagem para prover a aparência visual dos objetos. Este trabalho apresentou o conceito de equivalência visual, o qual considera que duas imagens são visualmente equivalentes se elas transmitem as mesmas impressões de aparência na cena. O modelo 3D da cena utilizado no experimento consistiu de uma esfera em quatro versões (uma com superfície suave e três versões com saliências), com propriedade de material representando alumínio escovado em quatro níveis, utilizando duas transformações no mapa de iluminação (*blurring* e *war-*

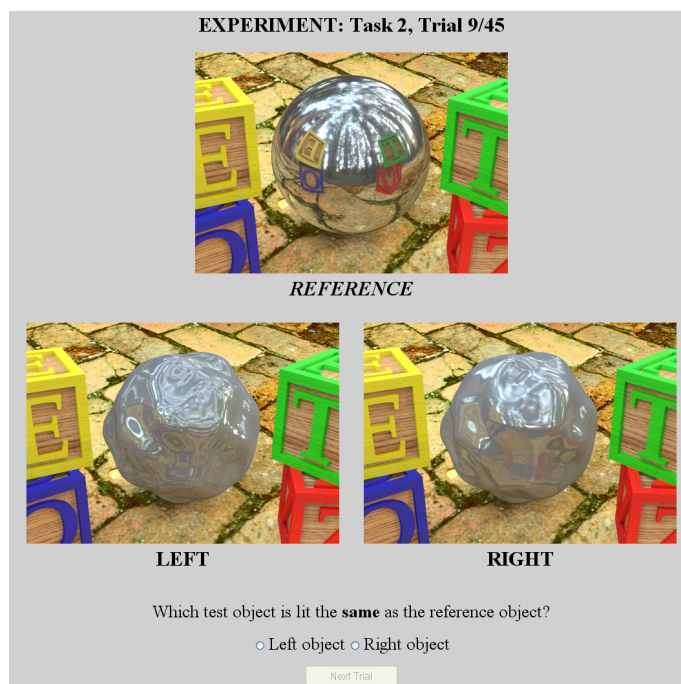


Figura 5 – Interface usada no experimento: imagem de referência posicionada ao topo e imagens de teste posicionadas abaixo. Fonte: (RAMANARAYANAN *et al.*, 2007).

ping). Uma imagem de referência foi posicionada ao topo da tela (Figura 5), enquanto duas imagens de teste seguem logo abaixo uma ao lado da outra (direita e esquerda). As duas imagens de teste apresentam o objeto em cena com mesma geometria e propriedade de material, a diferença está no mapa de iluminação: enquanto uma utiliza mesmo mapa da cena de referência, a outra utiliza um mapa com transformações. Os participantes responderam a perguntas comparando as imagens de teste com a imagem de referência, por exemplo quanto à forma ou material do objeto na cena. Os resultados dos testes foram utilizados para derivar uma métrica para prever a equivalência visual para as transformações nos mapas de iluminação.

Silva *et al.* (2008) desenvolveram um protocolo experimental para avaliar a qualidade visual perceptível de modelos 3D simplificados. Foram utilizados cinco modelos (Bunny, ChinaDog, Head, TRex e Hand), os quais diferem em número de faces e características da superfície, como detalhes pequenos e grandes regiões suaves. Para cada modelo 3D foram gerados seis níveis de simplificação utilizando três algoritmos (QSlim e dois outros providos pela biblioteca OpenMesh). O modelo 3D original foi posicionado à esquerda da tela, enquanto outras quatro versões simplificadas foram posicionadas à direita. Foi requisitado aos participantes a ordem de preferência dos quatro modelos 3D apresentados na tela (posição de 1 a 4 quanto a qualidade visual).

Os participantes podiam interagir livremente com os modelos 3D. Os dados coletados consideraram a ordem de preferência dos modelos, o número de interações e até mesmo o tempo de decisão para cada participante. Resultados mostraram, por exemplo, que observadores tendem a decidir mais rapidamente quanto mais simplificado é o modelo 3D. Os algoritmos de simplificação utilizados apresentaram comportamentos diferentes dependendo do modelo 3D e o nível de simplificação.

Bian, Hu e Martin (2009) realizaram um experimento para provar que a métrica de qualidade visual baseada na teoria de campos de energia produz resultados correlacionados com a opinião humana. Foram criadas variantes dos modelos 3D cavalo e rei do xadrez a partir da aplicação de marca d'água com o método *Triangle Similarity Quadruple* (OHBUCHI; MASUDA; AONO, 1998) e um método para adição de ruído. Ao todo foram criadas sete variantes com marca d'água mais sete variantes com ruído para cada modelo 3D. Os modelos 3D foram coloridos em cinza, apresentados em fundo preto, com modelo de iluminação global. Os participantes compararam o modelo 3D variante com o original e atribuíram um valor na escala de 0 (idêntico) a 10 (muito diferente).

Lavoué e Corsini (2010) realizaram um estudo comparativo entre as métricas baseadas na percepção humana, considerando diferentes cenários e mecanismos de percepção. As métricas estudadas foram: distância de Hausdorff, RMSE, MSDM (LAVOUÉ *et al.*, 2006), duas métricas de Corsini *et al.* (2007) e a métrica de Bian, Hu e Martin (2009). Foram realizados três experimentos, os quais são descritos na sequência.

O objetivo do primeiro experimento foi analisar se as métricas seguem o princípio de que distorções de alta frequência tem maior probabilidade de serem percebidas. Para isso, foram selecionados os modelos 3D Bimba e Dyno por serem bem diferentes entre si. O modelo 3D Bimba possui partes da superfície suaves enquanto o modelo 3D Dyno apresenta rugosidades em quase toda sua superfície. Para cada modelo 3D, foi produzida uma versão modificada de baixa frequência e outra de alta frequência obtida com adição de ruído aleatório. Foi constatado que as métricas baseadas na perceptividade têm bom comportamento em relação a esse princípio.

O objetivo do segundo experimento foi avaliar o comportamento das métricas em relação ao efeito de mascaramento visual, no qual um conteúdo de alta frequência

pode ser escondido por outro padrão. Foram utilizados quatro modelos 3D contendo tanto partes suaves quanto rugosas: Bimba e Dyno do experimento anterior, mais os modelos 3D Armadilho e Lion Head. Para cada modelo 3D foram produzidas versões modificadas com adição de ruído nas partes suaves (três versões) e depois apenas nas partes rugosas (três versões). O modelo 3D original mais as seis versões modificadas foram apresentadas na tela para os participantes, os quais atribuíram um valor entre 4 (idêntico) e 0 (muito diferente) para cada versão modificada em comparação ao modelo 3D original. Os participantes podiam interagir livremente com os modelos 3D. A métrica MSDM apresentou melhores resultados que as demais em relação ao mascaramento visual.

Já o objetivo do terceiro experimento foi avaliar as métricas em contexto geral, considerando várias distorções possíveis na malha. Os modelos 3D utilizados foram Armadillo, Dyno, Venus e RockerArm. Nesses modelos foram aplicadas dois tipos de distorções: adição de ruído e suavização, em três níveis (alto, médio, baixo) e em diferentes regiões da malha (em toda malha, regiões suaves, regiões rugosas e áreas intermediárias). Os participantes atribuíram valor de 0 (idêntico ao original) a 10 (pior caso). Assim como no experimento anterior, foi permitido aos participantes interagir livremente com o modelo 3D. A correlação das métricas com a percepção humana foi avaliada considerando apenas um tipo de distorção (suavização ou ruído) e ambos os tipos (suavização e ruído) para cada modelo 3D. A métrica MSDM e a primeira métrica de Corsini *et al.* (2007) apresentaram os resultados mais estáveis que as demais.

Os cenários dos experimentos apresentados nesta seção são resumidos nas Tabelas 2 e 3.

3.4 CONCLUSÃO

Neste capítulo foram apresentados os principais trabalhos relacionados à preservação digital e visualização de modelos 3D do patrimônio cultural. Esses trabalhos têm apresentado técnicas inovadoras para a digitalização de modelos 3D.

A maioria dos métodos para compressão 3D progressiva consideram modelos 3D constituídos apenas por geometria e conectividade. Poucos trabalhos apresentam soluções para modelos 3D coloridos (LEE; LAVOUÉ; DUPONT, 2010, 2012) ou texturizados (CAI; QI; SHEN, 2007). De acordo com a literatura científica pesquisada neste

trabalho, apenas Lee, Lavoué e Dupont (2012) abordaram a contribuição da preservação das cores na qualidade visual dos modelos intermediários. O método proposto por Cai, Qi e Shen (2007) considera que há uma imagem de textura para cada modelo intermediário, quando a solução mais apropriada é usar uma única imagem de textura para ser compartilhada por todos os modelos. Contudo, as soluções existentes de mapeamento de textura para modelos 3D progressivos (SANDER *et al.*, 2001; CHEN; CHUANG, 2006) não são apropriadas à compressão e visualização *web*, pois requerem espaço de armazenamento e memória adicionais.

Entre as métricas objetivas para avaliação da qualidade visual, a maioria calcula a similaridade entre duas superfícies baseada nas características geométricas. Apenas a métrica de Roy, Fofou e Truchetet (2004) possibilita a comparação entre dois modelos 3D baseada no atributo de cor. Para modelos 3D com textura, as métricas de Pan, Cheng e Basu (2005) e Tian e AlRegib (2004) mensuram a qualidade baseada na resolução da textura.

Experimentos subjetivos indicam que há benefícios em substituir geometria por textura, desde que esta tenha uma boa resolução (RUSHMEIER; ROGOWITZ; PIATKO, 2000). Os experimentos conduzidos por Pan, Cheng e Basu (2005) a fim de compreender como geometria e textura afetam a qualidade visual dos modelos 3D sugerem que observadores humanos são mais sensíveis a distorções de textura que de geometria.

Tabela 2 – Cenários dos experimentos subjetivos para a avaliação da qualidade visual de modelos 3D
- Parte I.

(a) Modelo 3D

Autor	Descrição	Número	Tamanho	Níveis de Simplificação
Rogowitz e Rushmeier (2001)	objetos bem conhecidos	2	69451 e 143206 tri-ângulos	2
Watson, Friedman e McGaffey (2001)	esfera	1	-	2
Rushmeier, Rogowitz e Piatko (2000)	objetos feitos pelo homem e representação de animais	36	370 polígonos	2
Pan, Cheng e Basu (2005)	objetos digitalizados	5	360x30 polígonos	6
Corsini <i>et al.</i> (2007)	objetos bem conhecidos	4	entre 50000 e 100000 triângulos	3
Ramanarayanan <i>et al.</i> (2007)	esfera	1	164000 vértices	0
Silva <i>et al.</i> (2008)	objetos diferentes em tamanho e características	5	entre 7932 e 34834 vértices	6
Bian, Hu e Martin (2009)	objetos diferentes em tamanho	2	12170 e 10024 triângulos	0
Lavoué e Corsini (2010)	diferentes naturezas e bem conhecidos	E2:5 E3:4	-	0

(b) Aparência do Modelo 3D

Autor	Cor/Textura/Material	Fonte de Luz/Iluminação	Sombreamento
Rogowitz e Rushmeier (2001)	colorido em cinza	ilumina de cima e de frente	-
Watson, Friedman e McGaffey (2001)	não	luz branca localizada no ponto de olho	flat shading
Rushmeier, Rogowitz e Piatko (2000)	textura	luz paralela e perpendicular à vista	gouraud shading (caso sem textura)
Pan, Cheng e Basu (2005)	textura	iluminação constante	-
Corsini <i>et al.</i> (2007)	efeito de pedra	único ponto de luz branca localizado no canto superior do objeto / iluminação local e difusa	-
Ramanarayanan <i>et al.</i> (2007)	alumínio escovado	2 mapas de iluminação (blurs e warps)	-
Silva <i>et al.</i> (2008)	-	-	-
Bian, Hu e Martin (2009)	colorido em cinza	única fonte de luz branca / iluminação global, difusa e especular	-
Lavoué e Corsini (2010)	efeito de pedra	única fonte de luz	gouraud e phong shading

- não mencionado

Tabela 3 – Cenários dos experimentos subjetivos para a avaliação da qualidade visual de modelos 3D - Parte II.

(a) Exibição

Autor	Plano de Fundo	Área da Tela	Monitor
Rogowitz e Rushmeier (2001)	preto*	-	-
Watson, Friedman e McGaffey (2001)	preto	591 em largura	17 polegadas CRT
Rushmeier, Rogowitz e Piatko (2000)	preto	370 em altura	1280x1024
Pan, Cheng e Basu (2005)	grade em verde representando o chão	750 em altura	21 polegadas CRT 1280x1024
Corsini <i>et al.</i> (2007)	degradê de azul para branco	600x600	17 polegadas 1280x600
Ramanarayanan <i>et al.</i> (2007)	dois cubos infantis em pátio de tijolos	-	20 polegadas LCD 1600x1200
Silva <i>et al.</i> (2008)	preto*	-	-
Bian, Hu e Martin (2009)	preto	preenche toda a tela	17 polegadas LCD 1280x1024
Lavoué e Corsini (2010)	-	-	-

(b) Participante

Autor	Número	Visualização/Interação	Modo de Avaliação
Rogowitz e Rushmeier (2001)	10	imagens + animações (rotação)	classificação 0..10
Watson, Friedman e McGaffey (2001)	36	imagens	tempo de nomeação, classificação e preferência
Rushmeier, Rogowitz e Piatko (2000)	8	imagens	classificação 0..100
Pan, Cheng e Basu (2005)	10	rotação (velocidade pode ser ajustada)	classificação 1..5
Corsini <i>et al.</i> (2007)	11	interação (rotação e escala)	classificação 1..10
Ramanarayanan <i>et al.</i> (2007)	30	imagens	perguntas sobre a diferença das imagens, forma, material e iluminação
Silva <i>et al.</i> (2008)	55	interação livre	ordem de preferência 1..4, ordem de apresentação na tela, número de interações e tempo de decisão
Bian, Hu e Martin (2009)	30	interação (rotação e escala)	classificação 0..10
Lavoué e Corsini (2010)	E2:11 E3:12	interação (rotação, escala, translação)	classificação E2:0..4 E3:0..10

- não mencionado

* cor não mencionada, porém pode ser vista nas figuras dos artigos

4 RECONSTRUÇÃO 3D PROGRESSIVA: A CONTRIBUIÇÃO DA PRESERVAÇÃO DAS CORES NA VISUALIZAÇÃO DE MODELOS 3D REALISTAS.

Este capítulo apresenta os experimentos realizados com o objetivo de analisar a contribuição da preservação das cores na visualização da reconstrução progressiva de modelos 3D realistas. Algoritmos para codificação da geometria, da conectividade e das cores são combinados para realizar a compressão progressiva dos modelos 3D coloridos. A partir desta combinação, duas abordagens de compressão foram testadas: (1) compressão tradicional e (2) compressão orientada por dados. Derivada da compressão tradicional, a compressão orientada por dados considera a importância visual das cores no processo de simplificação. Para a avaliação da qualidade visual dos modelos 3D comprimidos com ambas as abordagens são utilizadas duas métricas de qualidade. As próximas seções apresentam a compressão dos modelos 3D coloridos, as métricas de qualidade utilizadas, a metodologia para realização dos experimentos e os resultados obtidos.

4.1 COMPRESSÃO PROGRESSIVA DE MODELOS 3D COLORIDOS

Para a codificação da conectividade foi utilizado o algoritmo de Alliez e Desbrun (2001), escolhido por ser um dos mais eficientes e por resultar em modelos intermediários com boa qualidade visual. A cada iteração, o algoritmo simplifica um conjunto de vértices de acordo com a valência, combinando as etapas de conquista para dizimação e conquista para limpeza. A conquista para dizimação consiste na travessia em largura baseada em *patches* (Figura 6) para a dizimação de vértices com valência entre 3 e 6. De forma similar, vértices com valência 3 são dizimados na conquista para limpeza no caso de malhas regulares. Ao final, a codificação da conectividade resulta em uma sequência de códigos de valências (dados da codificação da conectividade).

Para a codificação da geometria foi utilizado o algoritmo proposto por Lee, Lavoué e Dupont (2009) que é um melhoramento do algoritmo de Alliez e Desbrun para codificação da geometria. Alliez e Desbrun primeiro aplicam uma quantização uniforme e global nas coordenadas dos vértices. Na dizimação, o residual dado pela diferença da posição do vértice removido com a média das posições dos seus vizinhos

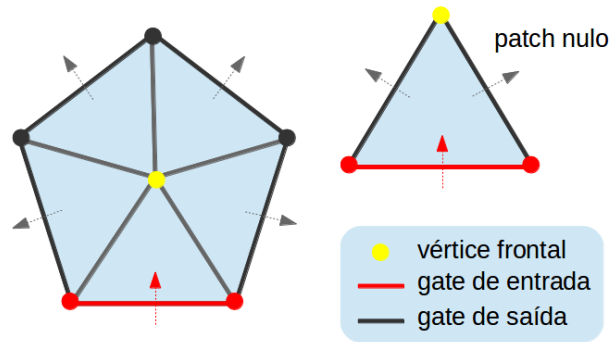


Figura 6 – Um *patch* consiste em um vértice (amarelo) e suas faces incidentes, possui um *gate* de entrada (vermelho) e *gates* de saída (cinza escuro) que levam aos *patches* adjacentes. *Gates* são arestas com direção, as quais são representadas por dois vértices (direita e esquerda) e uma face frontal. Esquerda: *patch* com valência 5. Direita: *patch* nulo.

(baricentro) é codificada em coordenadas Frenet Frame local. Lee *et al.* apresentaram uma otimização baseada em bijeção discreta para eliminar a etapa de pós-quantização do algoritmo de Alliez e Desbrun, etapa esta necessária para recuperar a posição exata do vértice. A codificação da geometria resulta em uma sequência de residuais codificados (dados da codificação da geometria).

Para a codificação das cores, outro algoritmo de Lee, Lavoué e Dupont (2010) para modelos 3D coloridos converte as componentes de cor RGB de cada vértice para o espaço CIE Lab. Na dizimação de um *patch*, o vértice com a cor mais próxima à média das cores dos vizinhos é selecionado. O residual da cor é dado então pela diferença da cor do vértice removido com a cor do vértice selecionado. Como resultado, tem-se uma sequência de residuais (dados da codificação da cor).

Baseado neste algoritmo para a codificação das cores, Lee, Lavoué e Dupont (2012) propuseram uma métrica que previne a dizimação de vértices que podem eventualmente causar distorção visual. A métrica para preservação das cores é composta pela combinação do erro de cor E_{cor} e o erro de geometria E_{geo} :

$$E_{total} = E_{cor} \otimes E_{geo} \quad (3)$$

Os autores assumem que o vértice a ser removido está localizado próximo do baricentro do *patch*. Assim, E_{cor} é dado pela diferença entre a cor do vértice a ser removido e a interpolação das cores dos vértices da face do meio gerada na retriangulação do *patch* (Figura 7). E_{geo} é dado pela razão entre a área média das faces

do *patch* (antes da dizimação) e a área média de todas as faces do modelo atual. O vértice então só é removido caso E_{total} seja menor que um limiar. Através da combinação dos erros de cor e geometria, a métrica possibilita a dizimação de vértices que, apesar da sua cor ser diferente das cores dos vizinhos, apresenta pequena influência em área.

Para ilustrar o funcionamento da métrica, a Figura 8 mostra os vértices de fronteira entre regiões com cores diferentes impedidos de serem removidos (coloridos em vermelho) na compressão do modelo 3D Monkey. Por guiar o processo de simplificação através do atributo de cor, esta é uma abordagem orientada por dados.

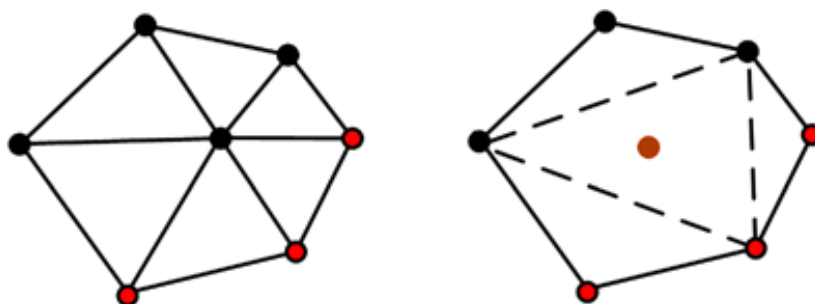


Figura 7 – Esquerda: vértice a ser removido em preto dentro do *patch*. Direita: cor marrom obtida na interpolação das cores dos vértices da face do meio. Fonte: (LEE; LAVOUÉ; DUPONT, 2012).

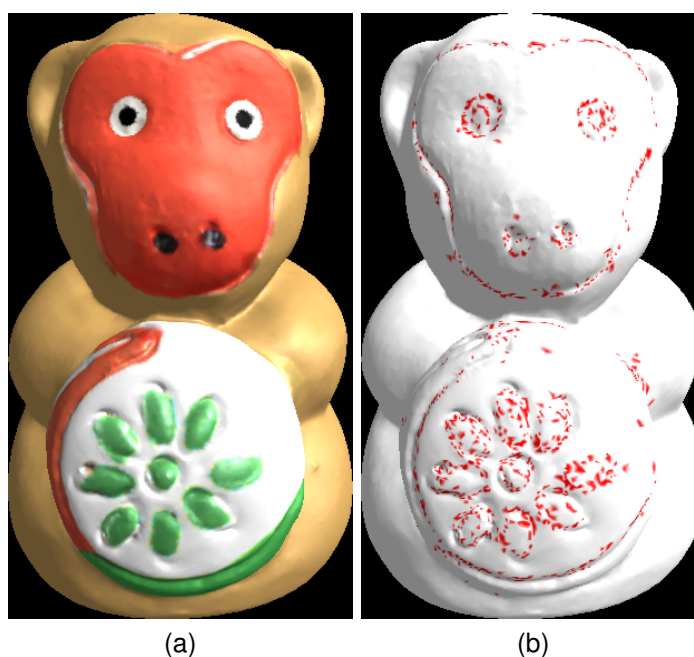


Figura 8 – Resultado visual da métrica proposta por Lee, Lavoué e Dupont (2012): (a) modelo 3D monkey e (b) vértices em vermelho impedidos de serem removidos durante a simplificação.

4.1.1 Codificação

O Algoritmo 1 mostra o pseudo-código da combinação dos algoritmos. Antes de iniciar cada iteração, é calculada a área média de todas as faces do modelo 3D atual, requisito para o cálculo do erro de geometria E_{geo} . Também, todos os vértices e faces são rotulados como “livre”. Um *gate* é escolhido arbitrariamente para iniciar a travessia, a qual é auxiliada por uma fila de *gates*¹. Dado um *gate* de entrada, são verificados os rótulos da face e do vértice frontal, além da valência deste vértice. Caso estejam livres e a valência seja entre 3 e 6, o *patch* pode ser dizimado desde que a remoção do vértice não cause distorção visual ($E_{total} < 0,5$ como definido em (LEE; LAVOUÉ; DUPONT, 2012)). Assim, o *patch* é deterministicamente retriangulado (detalhes da retriangulação em (ALLIEZ; DESBRUN, 2001)), gerando uma nova configuração. Caso contrário, a face frontal é definida como *patch* nulo. No cálculo do *patch* atual (linhas 11 e 22), vértices e faces do *patch* são rotulados como “conquistados”.

A codificação da conectividade (linhas 14 e 23) gera um código de valência do vértice removido (3, 4, 5 ou 6) ou um código de *patch* nulo (0, 1, 2 ou 7 podem ser utilizados). Na codificação da geometria (linha 15), o residual dado pela diferença da posição do vértice frontal e o baricentro do *patch* é projetado para o Frenet Frame local usando a bijeção discreta proposta em (LEE; LAVOUÉ; DUPONT, 2009). Na codificação da cor (linha 16), o residual de cor é dado pela diferença da cor do vértice frontal com a cor do vértice mais próxima à média das cores dos vértices do *patch*, como proposto em (LEE; LAVOUÉ; DUPONT, 2010). Na sequência, os *gates* de saída do *patch* atual são adicionados à fila. A conquista para dizimação termina quando não houver mais *gates* na fila. Cada iteração do algoritmo tem como resultado final uma versão simplificada do modelo 3D de entrada (modelo intermediário) mais os dados obtidos na codificação da conectividade, da geometria e da cor.

4.1.2 Decodificação

Para a decodificação de cada modelo intermediário, é feita a mesma travessia do Algoritmo 1 com o objetivo de encontrar exatamente os mesmos *patches* conquistados. Essa travessia deve ser iniciada com o mesmo *gate* usado para iniciar a conquista para dizimação (*gate_inicial*). A partir de um *gate* de entrada, o rótulo da face frontal

¹Estrutura de dados do tipo fila FIFO (First In, First Out).

e o código de valência são verificados. Caso a face frontal esteja livre e o código não seja nulo, as bordas do *patch* são descobertas. Para um código de valência v , a descoberta requer caminhar por $(v - 2)$ faces livres, caminho este determinado pela forma como o *patch* foi retriangulado. Face frontal livre e código de valência atual nulo significa que a face é um *patch* nulo. Recuperam-se as coordenadas e as componentes de cor do novo vértice somando-se os residuais aos valores predizíveis (baricentro e cor mais próxima respectivamente). Por fim, o novo vértice é inserido no *patch*, seguido por retriangulação.

Algoritmo 1 Pseudo-código do algoritmo de conquista para dizimação.

Requer: Cálculo da área média de todas as faces do modelo atual

Requer: Todos os vértices e faces rotulados como “livre”

Requer: *gate_inicial*

```

1: Insira gate_inicial na fila
2: enquanto houver gates na fila faça
3:   gate_atual ← remova um gate da fila
4:   vertice_frontal ← calcule o vértice frontal a partir do gate_atual
5:   valencia ← calcule a valência do vertice_frontal
6:   se a face frontal do gate_atual está rotulada como “conquistado” ou o
     vertice_frontal está rotulado como “conquistado” então
7:     // nada a ser feito
8:   senão
9:     patch_nulo ← verdadeiro
10:    se vertice_frontal está rotulado como “livre” e sua valência é entre 3 e 6 então
11:      patch_atual ← calcula_patch(vertice_frontal)
12:      erro_total ← calcula_distorcao_cor(patch_atual)
13:      se erro_total < 0,5 então
14:        codifica_conectividade(valencia)
15:        codifica_geometria(patch_atual)
16:        codifica_cor(patch_atual)
17:        Insira os gates de saída do patch_atual na fila
18:        patch_nulo ← falso
19:      fim se
20:    fim se
21:    se patch_nulo == verdadeiro então
22:      patch_atual ← calcula_patch(gate_atual)
23:      codifica_conectividade(0)
24:      Insira os gates de saída do patch_atual na fila
25:    fim se
26:  fim se
27: fim enquanto

```

4.2 MÉTRICAS PARA A AVALIAÇÃO DA QUALIDADE VISUAL DE MODELOS 3D COLORIDOS

Para a avaliação da qualidade visual dos modelos intermediários foram utilizadas duas abordagens de RMSE existentes as quais podem ser aplicadas a modelos 3D coloridos. A métrica de Roy, Foufou e Truchetet (2004) mensura a distorção entre dois modelos 3D baseada na diferença entre atributos de vértices. O *desvio atributo* entre um vértice v e uma superfície S é definida como a distância do atributo i do vértice v ao atributo i do vértice mais próximo em S :

$$d_i(v, S) = ||f_i(v) - f_i(N_S(v))||, \quad (4)$$

tal que $N_S(v)$ é igual ao vértice mais próximo. Dados dois modelos 3D A (original) e B (intermediário) e suas respectivas superfícies S_A e S_B e um conjunto de vértices $\{v_j | v_j \in S_A \text{ e } j = 1, 2, \dots, m\}$, o RMSE baseado no atributo de cor é dado por:

$$RMSE_{Roy}(A, B) = \sqrt{\frac{1}{m} \sum_{j=1}^m (d_{cor}(v_j, S_B))^2} \quad (5)$$

Baseado no cálculo do RMSE para duas imagens (Equação 2), Lindstrom e Turk (2000) propuseram combinar múltiplas vistas do modelo 3D em uma única medida de erro. Para obter uma boa cobertura, a câmera é posicionada em torno do modelo 3D utilizando os vértices de um pequeno *rhombicuboctahedron*, o qual contém 24 vértices equidistantes. Dados 24 *snapshots*² com resolução 512x512 *pixels* dos modelos 3D X (original) e Y (intermediário), o RMSE é dado por:

$$RMSE_{LT}(X, Y) = \sqrt{\frac{1}{24 * 512 * 512} \sum_{k=1}^{24} \sum_{j=1}^{512} \sum_{i=1}^{512} (x_{ijk} - y_{ijk})^2} \quad (6)$$

4.3 METODOLOGIA

A partir do Algoritmo 1 são obtidas duas abordagens baseadas no método de Alliez e Desbrun (2001): Lee, Lavoué e Dupont (2010) para codificação do atributo de cor (LLD) e uma versão desta adicionando a métrica para a preservação das cores

²Imagens de luminância capturadas da tela de visualização do modelo 3D.

proposta em seu trabalho posterior (LEE; LAVOUÉ; DUPONT, 2012) (LLD + métrica cor). Assim, as abordagens se diferenciam quanto à verificação da distorção de cor. LLD realiza a compressão tradicional e não executa as linhas 12 e 13 do algoritmo, enquanto LLD + métrica cor realiza a compressão orientada por dados.

Os experimentos foram conduzidos para 4 modelos 3D do acervo do IMAGO gerados para preservação digital: os objetos indígenas Anta e Índias, e as esculturas dos profetas Joel e Baruc, feitas por Aleijadinho. Os objetos indígenas são pequenos, enquanto as esculturas dos profetas possuem tamanho próximo ao natural (altura de uma pessoa). Assim, os modelos 3D dos profetas possuem aproximadamente 2-3M vértices, já os modelos 3D indígenas são menores (aproximadamente 150-200K vértices). Cada modelo 3D foi comprimido progressivamente com as duas abordagens de compressão.

Para a avaliação da qualidade visual da reconstrução 3D progressiva, calculou-se o RMSE com a métrica de Roy, Fofou e Truchetet (2004) de cada modelo intermediário em relação ao modelo 3D original. Os *snapshots* para o cálculo do RMSE para uma vista ou para o RMSE de Lindstrom e Turk (2000) foram obtidos através da renderização dos modelos utilizando os mesmos parâmetros de visualização. Os resultados obtidos com as métricas de qualidade são apresentados através de curvas de distorção visual. São selecionados dois modelos intermediários com similar RMSE na métrica M_1 . Estes dois modelos são marcados nas curvas de distorção visual da métrica M_2 . Assim, as métricas são avaliadas quanto à correlação aos resultados visuais da renderização dos modelos selecionados. As vistas dos objetos que mostram os resultados visuais são escolhidas através de análise visual criteriosa, de forma a mostrar as principais similaridades ou diferenças entre as abordagens de compressão.

Os experimentos foram realizados usando um computador com a seguinte configuração: Intel(R) Core(TM)2 Quad Processor Q8200 (2.33 GHz), 4GB de memória e NVIDIA GeForce GTX 560.

4.4 RESULTADOS EXPERIMENTAIS

A Tabela 4 apresenta as taxas de *bits* requeridas para a reconstrução da conectividade do modelo original (A) a partir do modelo base (B) para as duas abordagens de compressão. A capacidade de compressão da conectividade para a aborda-

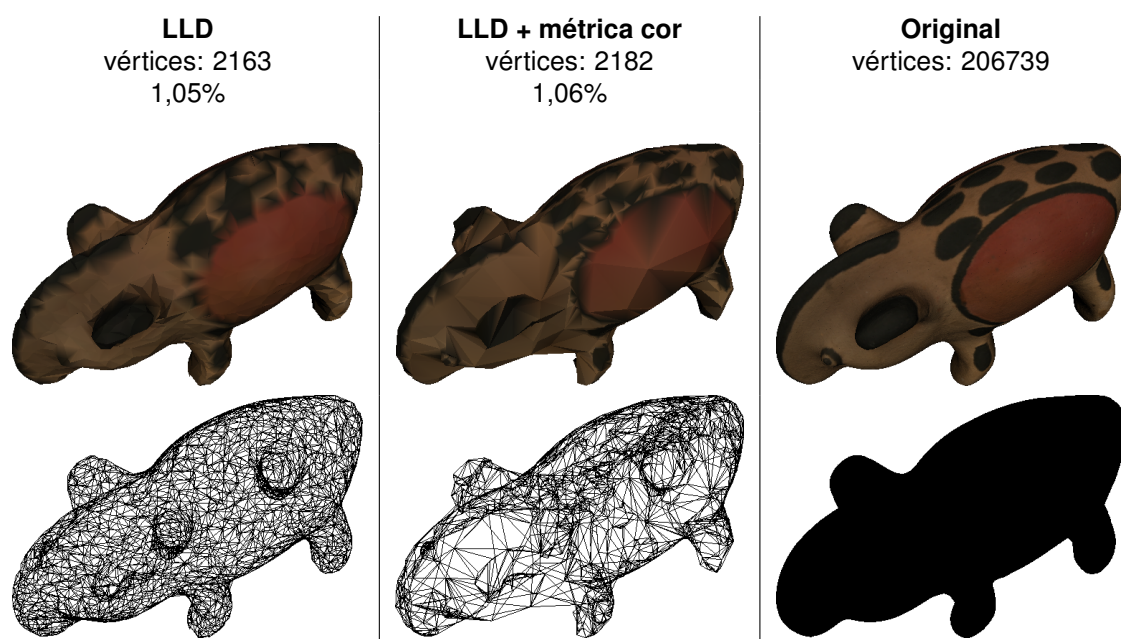
gem LLD + métrica cor está diretamente relacionada às penalidades que são causadas ao impedir a dizimação de um vértice. Essa restrição gera um código de *patch* nulo, causando uma penalidade à taxa de compressão da conectividade. Considerando que cada código requer 3 *bits* para armazenar uma valência, são 3 *bits* que não adicionam novos elementos à malha. Para minimizar as penalidades, a métrica cor possibilita a dizimação de vértices que, apesar da sua cor ser diferente das cores dos vizinhos, apresentam pequena influência em área (através da combinação dos erros de cor e geometria). No entanto, para os modelos 3D coloridos testados neste experimento, os quais possuem alta resolução, a capacidade de compressão da conectividade sofreu impacto considerável. LLD + métrica cor requer mais que o dobro de *bits* para reconstruir a conectividade dos modelos 3D dos profetas em relação à versão sem a métrica.

Tabela 4 – Taxas de compressão dos modelos testados em *bits* por vértice. Quantização usando 10 *bits*.

(a) Índias			
Abordagem	#vértices $A \rightarrow B$	Iterações	Conectividade (bit/v)
LLD	148338 \rightarrow 15700	6	8,409
LLD + métrica cor	148338 \rightarrow 14984	10	14,943
(b) Anta			
Abordagem	#vértices $A \rightarrow B$	Iterações	Conectividade (bit/v)
LLD	206739 \rightarrow 2163	12	8,282
LLD + métrica cor	206739 \rightarrow 2182	26	13,841
(c) Joel			
Abordagem	#vértices $A \rightarrow B$	Iterações	Conectividade (bit/v)
LLD	1974038 \rightarrow 22844	12	8,390
LLD + métrica cor	1974038 \rightarrow 23304	32	18,760
(d) Baruc			
Abordagem	#vértices $A \rightarrow B$	Iterações	Conectividade (bit/v)
LLD	2887390 \rightarrow 67772	10	8,598
LLD + métrica cor	2887390 \rightarrow 66784	21	17,034

A utilização da métrica cor requer mais iterações do algoritmo de compressão a fim de alcançar um modelo base altamente simplificado. A Tabela 5 mostra os resultados visuais do modelo 3D Anta obtidos com as abordagens de compressão. Os modelos base apresentados possuem similar número de vértices (aproximadamente 1,5% de vértices do modelo 3D original). Observa-se em LLD + métrica cor a preser-

Tabela 5 – Diferenças visuais entre as abordagens de compressão para modelos base com número similar de vértices. Renderização do modelo 3D Anta com cores e renderização em *wireframe*.



vação das regiões delimitadas por descontinuidade de cor, como as manchas pretas nas costas e o contorno da região vermelha na lateral da Anta. Com tal característica, LLD + métrica cor apresenta uma superfície com regiões mais simplificadas do que outras, enquanto LLD apresenta uma superfície suave, como se pode observar na renderização em *wireframe*.

Os gráficos da Figura 9 mostram as curvas de distorção visual obtidas através do RMSE de Roy e do RMSE para uma vista do modelo 3D Joel, a qual mostra simultaneamente a face e o pergaminho do profeta. As curvas ilustram os valores RMSE em função da porcentagem de elementos (vértices e faces) decodificados na reconstrução 3D progressiva. Os valores RMSE são normalizados para facilitar as comparações. Dois modelos intermediários com similar RMSE em Roy foram selecionados (15,44% de LLD + métrica cor e 32,95% de LLD marcados com ×) e marcados nas curvas de distorção visual para uma vista.

Nas figuras da Tabela 6 pode-se observar a similaridade visual entre as abordagens de compressão em relação às cores, as quais contribuem significativamente para a leitura das passagens bíblicas do pergaminho. Já na região da face, observa-se menor distorção geométrica na abordagem LLD. Essas diferenças são percebidas com ampliação ou se os modelos são comparados ao modelo 3D original.

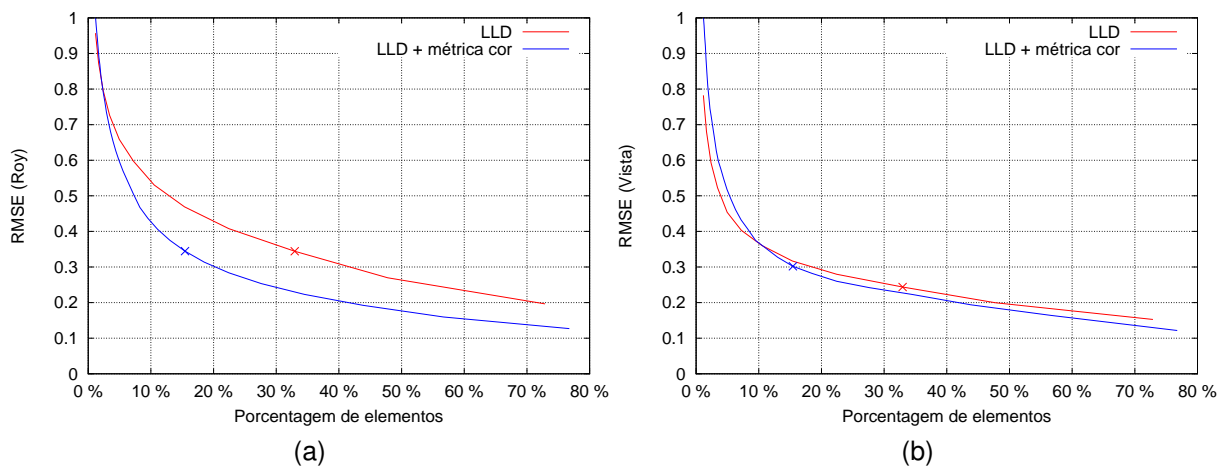
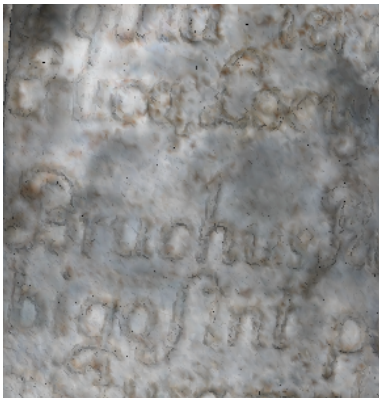
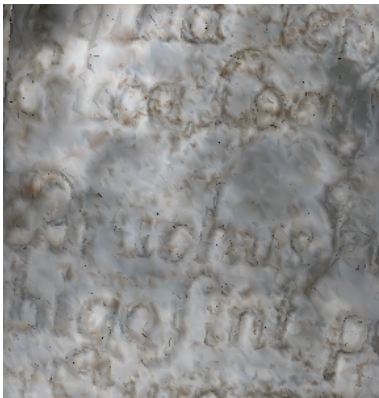






Figura 9 – Avaliação da qualidade visual do modelo 3D Joel para as abordagens de compressão. Curvas de distorção visual obtidas através do RMSE de Roy em (a) e do RMSE para uma vista em (b).

Tabela 6 – Avaliação da qualidade visual do modelo 3D Joel para as abordagens de compressão. Comparação visual dos modelos intermediários com similar RMSE com o modelo 3D original.

LLD (×) vértices: 650383 32,95%	LLD + métrica cor (×) vértices: 304713 15,44%	Original vértices: 1974038
		
		

As curvas de Roy e as curvas para uma vista apresentam resultados diferentes. Em Roy, as abordagens de compressão possuem similar RMSE até aproximadamente 5% de elementos. Após esta porcentagem de elementos, LLD + métrica cor tem maior queda de distorção visual. Significa que a partir deste ponto, a preservação das cores através da métrica cor contribui para melhorar a qualidade visual do modelo 3D. Analisando as curvas para uma vista, LLD + métrica cor supera LLD aproximadamente após 10% de elementos, porém os valores RMSE são próximos. Apesar dessas diferenças, as curvas apresentam uma característica semelhante: queda acentuada nos primeiros níveis seguida por estabilização.

As curvas de distorção visual do modelo 3D Anta são apresentadas na Figura 10. Foram selecionados dois modelos intermediários com similar RMSE em Roy (39,43% de LLD + métrica cor e 72,55% de LLD marcados com ×). A vista da lateral da Anta foi selecionada para o cálculo do RMSE para uma vista. De acordo com as curvas para uma vista, em torno de 1,5% de elementos (equivalente aos modelos base apresentados na Tabela 5) LLD apresenta menor valor RMSE que LLD + métrica cor, enquanto as curvas de Roy indicam o contrário. Apesar desta diferença, as curvas de Roy e as curvas para uma vista mostram que há queda acentuada nos primeiros níveis seguida por estabilização. Nas figuras da Tabela 7 pode-se observar a forte similaridade visual entre as abordagens de compressão e o modelo 3D original.

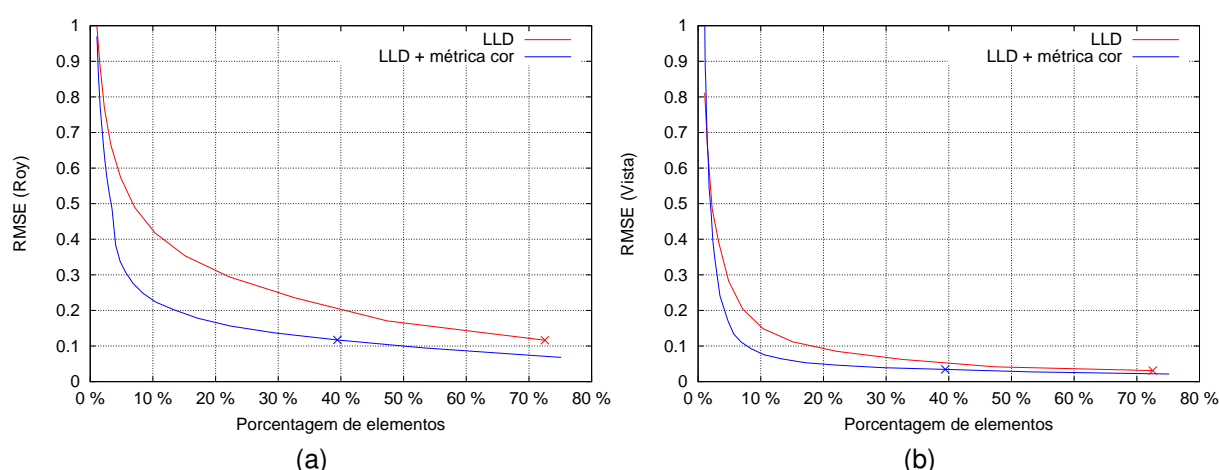
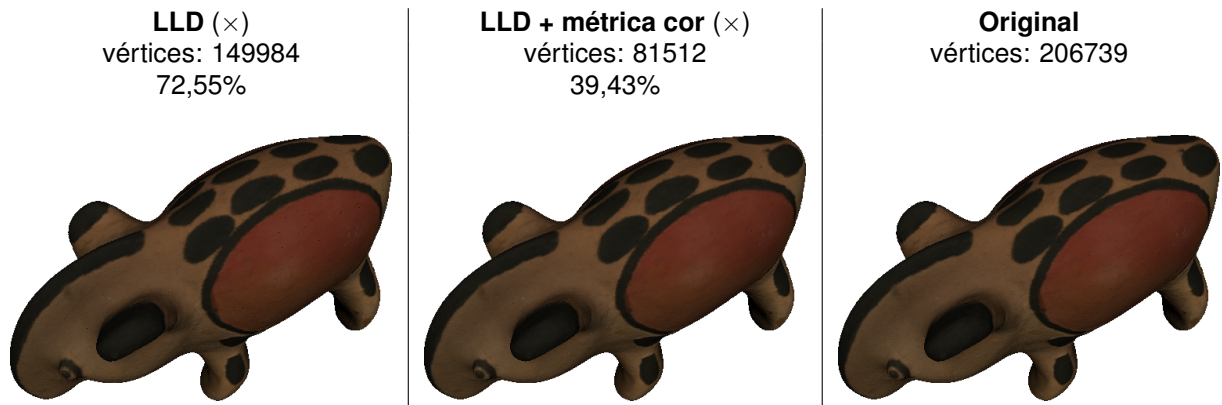


Figura 10 – Avaliação da qualidade visual do modelo 3D Anta para as abordagens de compressão. Curvas de distorção visual obtidas através do RMSE de Roy em (a) e do RMSE para uma vista em (b).

Tabela 7 – Avaliação da qualidade visual do modelo 3D Anta para as abordagens de compressão. Comparação visual dos modelos intermediários com similar RMSE com o modelo 3D original.



Já que uma única vista pode não capturar toda a aparência do objeto, calculou-se o RMSE para um conjunto de vistas como proposto por Lindstrom e Turk (LT). A Figura 11 apresenta as curvas de distorção visual do modelo 3D Índias, o menor dos modelos em resolução. As curvas de Roy e as curvas de LT mostram comportamento semelhante: LLD + métrica cor supera LLD para todos os modelos intermediários. Foram selecionados dois modelos intermediários com similar RMSE em Roy (42,03% de LLD + métrica cor e 72,86% de LLD marcados com \times). As figuras da Tabela 8 apresentam os resultados visuais. Pode-se observar a similaridade visual entre as abordagens de compressão, apesar do modelo intermediário LDD + métrica cor possuir menor número de elementos.

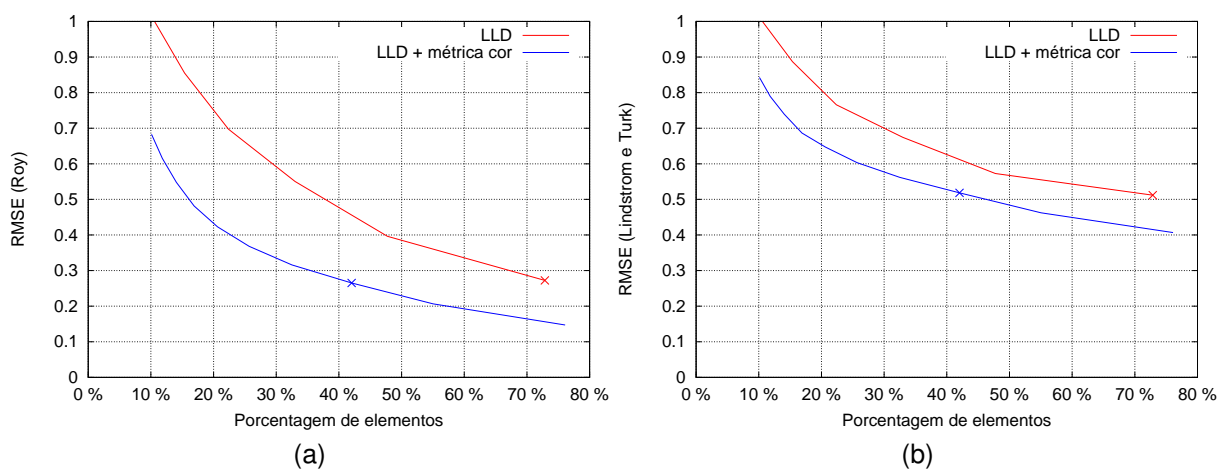


Figura 11 – Avaliação da qualidade visual do modelo 3D Índias para as abordagens de compressão. Curvas de distorção visual obtidas através do RMSE de Lindstrom e Turk em (a) e do RMSE de Roy em (b).

Tabela 8 – Avaliação da qualidade visual do modelo 3D Índias para as abordagens de compressão. Comparação visual dos modelos intermediários com similar RMSE.



Esta mesma característica não é observada para o modelo 3D Baruc, o maior dos modelos em resolução. A Figura 12 apresenta as curvas de distorção visual. As curvas de LT indicam superioridade da abordagem LLD + métrica cor após aproximadamente 25% de elementos. Foram selecionados dois modelos intermediários com similar RMSE em LT antes (2,35% LLD e 8,58% LLD + métrica cor marcados com □) e após (26,59% LLD + métrica cor e 32,99% LLD marcados com ×) 25% de elementos. Os resultados visuais são apresentados na Tabela 9. Analisando os dois modelos que antecedem o ponto de interseção, a certa distância eles parecem similares. No entanto, ao ampliar a região do pergaminho, observa-se que as passagens bíblicas são mais nítidas em LLD + métrica cor. Diferente das curvas de LT, as curvas de Roy indicam superioridade de LLD + métrica cor para todos os modelos intermediários. Analisando os dois modelos após o ponto de interseção, observa-se a similaridade visual entre as abordagens de compressão, enquanto Roy indica que LLD + métrica cor possui menor distorção visual.

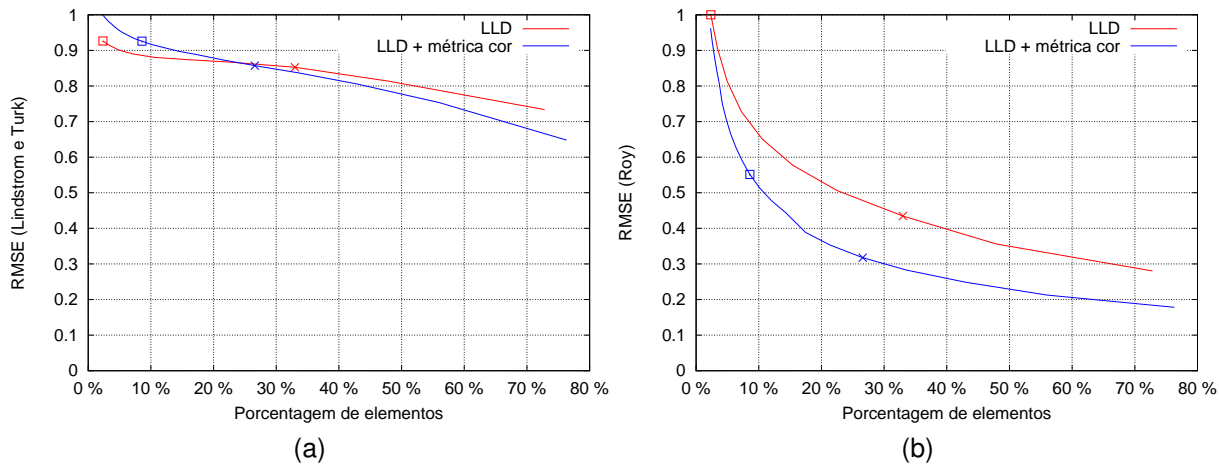
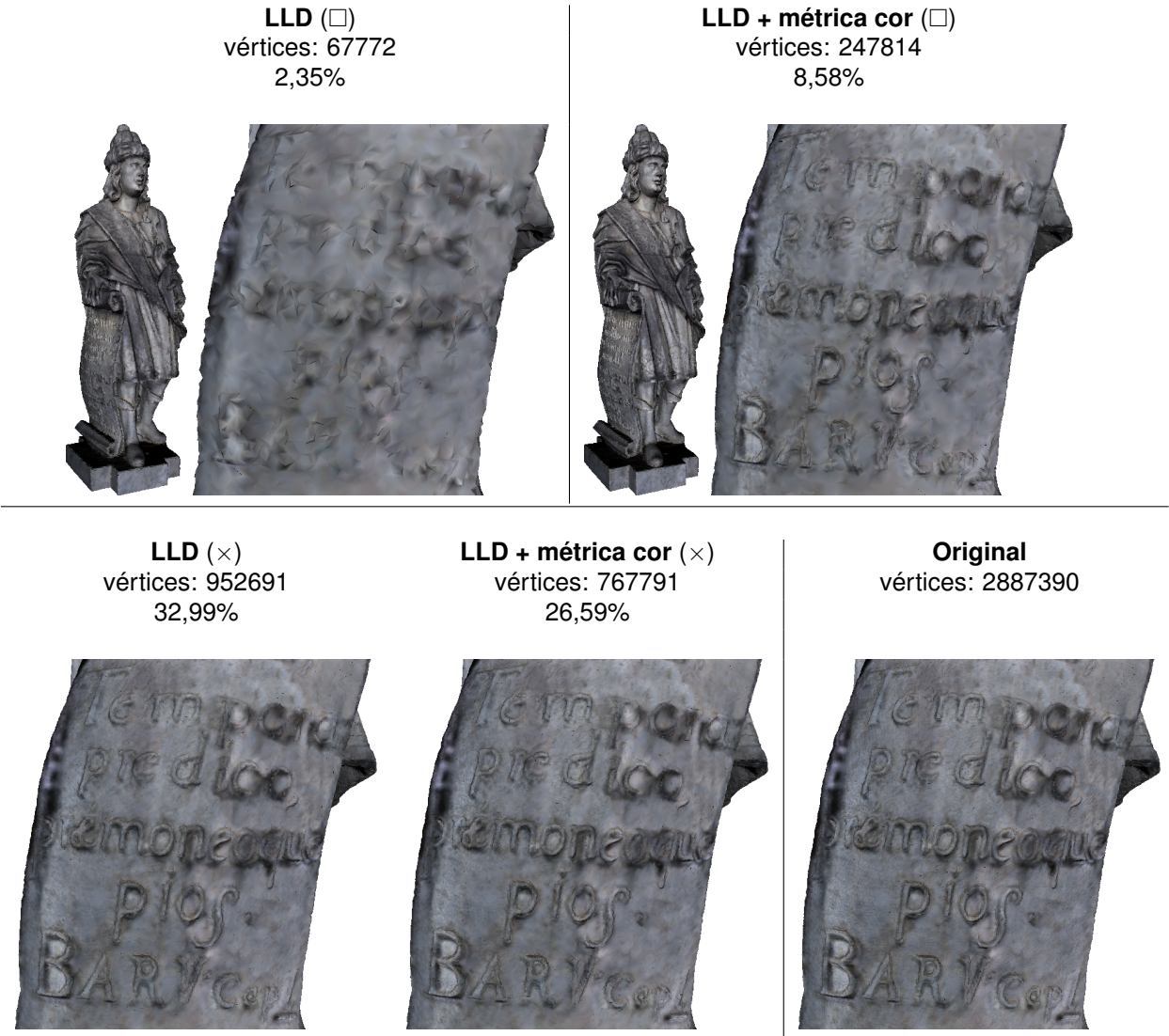


Figura 12 – Avaliação da qualidade visual do modelo 3D Baruc para as abordagens de compressão. Curvas de distorção visual obtidas através do RMSE de Lindstrom e Turk em (a) e do RMSE de Roy em (b).

Tabela 9 – Avaliação da qualidade visual do modelo 3D Baruc para as abordagens de compressão. Comparação visual dos modelos intermediários com similar RMSE.



4.5 DISCUSSÃO DOS RESULTADOS E CONCLUSÃO

As curvas de Roy e as curvas para uma vista do modelo 3D Joel apresentam resultados diferentes. Os resultados para uma única vista são similares aos de Roy apenas para modelos menores em resolução e com superfícies menos detalhadas (*e.g.* Anta). Para o modelo 3D Índias, menor em resolução entre os modelos testados, a combinação de 24 vistas conseguiu capturar a aparência do objeto. Os resultados obtidos com Roy e com LT para este modelo são similares. O mesmo não foi observado para o modelo 3D Baruc. Visualizados a distância, os dois modelos intermediários similares em Roy antes do ponto de interseção das curvas são parecidos. No entanto, aos serem ampliados, as diferenças são facilmente percebidas. Além de maiores em resolução, os modelos dos profetas possuem superfícies com mais detalhes, o que requer um conjunto maior de vistas para capturar toda a aparência do objeto.

Na comparação visual dos quatro modelos 3D testados, é possível observar que a partir de uma determinada resolução os modelos intermediários apresentam boa aproximação ao modelo 3D original. Pequenas diferenças são observadas apenas com ampliação (*e.g.* modelo 3D Joel na Tabela 6), ou praticamente não são percebidas (*e.g.* pergaminho do modelo 3D Baruc na Tabela 9). Para o modelo 3D Baruc, os dois modelos intermediários em LT após a interseção das curvas são visualmente similares, provavelmente devido à alta resolução que esses modelos possuem. No entanto, a métrica de Roy indica diferença considerável entre esses dois modelos, mesmo sendo visualmente indistinguíveis.

Apesar de combinar cor e geometria em seu cálculo, a métrica de Roy apresenta resultados mais correlacionados à análise visual quanto à preservação das cores do que em relação à geometria. Por exemplo para o modelo 3D Anta, a métrica de Roy indica pequena superioridade de LLD + métrica cor para os modelos base (aproximadamente 1,5% de elementos), como é possível observar nas figuras da Tabela 7 (destaque para a preservação do contorno da região vermelha na lateral da Anta). No entanto, um observador pode julgar o modelo base LLD visualmente melhor ao considerar a topologia, como indicam os valores RMSE para uma vista.

Quanto à contribuição da preservação das cores, os resultados dos experimentos são consistentes. A preservação das cores através da métrica cor é capaz

de gerar modelos intermediários com menos elementos, porém visualmente similares aos gerados sem a métrica. Por exemplo, um modelo LLD pode ser substituído por um modelo LLD + métrica cor. Mais do que isso, um modelo LLD + métrica cor pode substituir o modelo 3D original. Isso possibilita interromper a reconstrução 3D progressiva em um determinado nível, por exemplo quando há restrições de acesso remoto. O desafio consiste em definir qual é este nível.

A métrica de Roy baseada no atributo de cor apresenta um modo eficiente para comparar dois modelos 3D coloridos, porém ela pode não indicar correta similaridade de acordo com a percepção visual humana. Roy pode indicar modelos similares quanto às cores, porém visualmente diferentes geometricamente, ou pode indicar que são diferentes, mesmo sendo visualmente parecidos. Confirmando experimentos previamente realizados (LAVOUÉ; CORSINI, 2010) para modelos 3D não coloridos, métricas objetivas de qualidade baseadas em distância euclidiana vértice a vértice não são correlacionadas à percepção visual humana. Portanto, também é difícil comparar objetivamente dois modelos 3D coloridos.

A preservação das cores mostra-se uma eficiente estratégia para a visualização de modelos 3D realistas na reconstrução 3D progressiva. A métrica cor apresenta bom funcionamento para preservação das cores, porém possui duas desvantagens: baixas taxas de compressão da conectividade e possível deformação da topologia a baixas resoluções. Apesar da métrica cor considerar o impacto da dizimação dos vértices para minimizar as penalidades, a abordagem requer mais *bits* para reconstruir a conectividade do modelo 3D em comparação à versão sem a métrica. Para alcançar um modelo base altamente simplificado, são necessárias mais iterações do algoritmo de compressão. A cada iteração, vértices de fronteira de regiões com cores diferentes são impedidos de serem dizimados, enquanto internamente as regiões são simplificadas cada vez mais, o que pode modificar a topologia do objeto.

Em geral, as curvas de distorção visual apresentam uma semelhança: queda acentuada nos primeiros níveis seguida por estabilização. Esta característica pode sugerir um ponto de degradação mínima para o modelo base. Como descrito no Capítulo 2, a compressão 3D progressiva busca o equilíbrio entre a taxa de compressão e a qualidade visual da aproximação. Portanto, um modelo base a partir do ponto de estabilização pode ser visualmente agradável, porém pode não ser pequeno o suficiente

para diminuir a latência de visualização, principalmente para grandes modelos 3D. O desafio consiste em preservar as cores e diminuir a distorção geométrica a baixas resoluções, ou seja, aproximar o início da curva ao ponto de estabilização.

Por fim, algumas considerações finais podem ser resumidas:

- O correto funcionamento do RMSE baseado em imagem depende da seleção das vistas. Uma vista ótima pode capturar a aparência do objeto, desde que a forma do objeto seja simples. Objetos que possuem forma complexa requerem um conjunto maior de vistas.
- O RMSE de Roy é eficiente para calcular a similaridade entre dois modelos 3D coloridos, apesar de demonstrar maior precisão na comparação das cores do que da geometria.
- A estabilização das curvas de distorção visual pode sugerir um ponto de degradação mínima para a compressão dos modelos 3D.
- A métrica cor preserva as cores no processo de simplificação, porém as restrições impostas à dizimação dos vértices pode impactar negativamente a capacidade de compressão da conectividade ou causar deformações na topologia a baixas resoluções.
- A preservação das cores contribui para melhorar a qualidade visual dos modelos 3D realistas na reconstrução 3D progressiva. O modelo 3D original pode ser substituído por versões geometricamente menos precisas porém visualmente similares.

Os resultados e constatações apresentados neste capítulo motivaram as soluções propostas nesta tese, como por exemplo uma nova abordagem de compressão 3D progressiva orientada por dados para modelos 3D coloridos descrita no Capítulo 5. Além de confirmar a contribuição da preservação das cores, os experimentos possibilitam um melhor entendimento sobre os desafios na avaliação da qualidade visual de modelos 3D coloridos.

5 COMPRESSÃO 3D PROGRESSIVA ORIENTADA POR DADOS

Devido às recentes tecnologias para digitalização 3D, os modelos 3D estão cada vez mais detalhados. Esses modelos são representados por uma grande quantidade de vértices, faces e atributos de aparência, como cores ou coordenadas de texturas. Nesse contexto, a compressão 3D progressiva reduz o espaço de armazenamento e melhora a latência, uma vez que permite a visualização do modelo 3D enquanto ele está sendo carregado. Como descrito no Capítulo 2, na compressão progressiva um modelo 3D de entrada é simplificado iterativamente até obter um modelo base. Nesse processo são gerados modelos intermediários, os quais podem ser visualizados na reconstrução 3D progressiva.

A maioria dos métodos de compressão progressiva consideram modelos 3D constituídos por geometria e conectividade. Trabalhos recentes focam na melhoria da taxa de compressão através da otimização da quantização. No entanto, são poucos os métodos para modelos 3D coloridos. As cores também podem ser representadas por meio de texturas coloridas, porém, pelo nosso melhor conhecimento, não há uma solução para o mapeamento de texturas para compressão 3D progressiva.

As cores contribuem para a interpretação dos objetos. Aplicações onde a qualidade visual também é um requisito importante, como por exemplo na visualização de modelos 3D da preservação digital, as cores fornecem informações relevantes sobre os objetos. Por exemplo, artefatos indígenas apresentam desenhos e cores que caracterizam cada tribo e sua arte. Regiões delimitadas por descontinuidade de cor são visualmente significativas, portanto sua preservação é essencial para a visualização de aproximações com qualidade.

Técnicas de mapeamento de textura são amplamente utilizadas para aumentar o realismo dos modelos 3D. O mapeamento de textura é uma maneira eficaz de adicionar detalhes à superfície quando há restrições ao uso de geometria. Nesse contexto, este capítulo apresenta uma abordagem de compressão progressiva de modelos 3D coloridos orientada por dados com o objetivo de aumentar a qualidade visual dos modelos intermediários a baixas resoluções. A ideia é representar as cores do modelo por meio de uma textura colorida ao invés de usar atributo de cor. O método possui quatro características importantes:

1. **Uma única textura colorida é compartilhada por todos os modelos intermediários.** Ao invés de armazenar uma textura colorida para cada modelo intermediário, a abordagem proposta requer apenas uma textura colorida.
2. **Representação compacta das coordenadas de textura.** Em geral, coordenadas de textura (v, t) são representadas utilizando 8 *bytes float*. A abordagem proposta utiliza apenas uma pequena textura contendo os dados necessários para gerar as coordenadas de textura de todos os modelos intermediários.
3. **Qualidade Visual.** Os modelos intermediários apresentam qualidade visual superior ao uso de atributo de cor em relação à preservação das cores a baixas resoluções.
4. Todas as contribuições citadas acima são aplicáveis e podem ser adaptadas para outros métodos progressivos.

5.1 VISÃO GERAL

A abordagem proposta é baseada em dois métodos: compressão 3D progressiva de Alliez e Desbrun (2001) e mapeamento de texturas por complexos de células cúbicas de Apaza-Agüero, Silva e Bellon (2011). Detalhado previamente no Capítulo 4, o método de compressão progressiva de Alliez e Desbrun é um dos mais eficientes orientados à conectividade, os quais produzem modelos intermediários com boa qualidade visual.

O método de Apaza-Agüero, Silva e Bellon (2011) tem como objetivo mapear uma textura contendo as cores do modelo em alta resolução para ser utilizada por uma versão simplificada desse modelo. O método resulta em duas texturas: uma textura colorida e uma textura pequena a partir da qual são geradas as coordenadas de textura durante a visualização baseada em GPU. Dessa maneira, este método é mais adequado ao contexto da compressão uma vez que usa uma estrutura de dados compacta para representar as coordenadas de textura.

O diagrama da Figura 13 apresenta o fluxograma da abordagem proposta. A partir do modelo 3D de entrada, são geradas as duas texturas: a pequena *textura de células* contendo a estrutura de células cúbicas e a *textura colorida* contendo as cores do modelo de entrada. A textura de células é usada na compressão 3D progressiva

para guiar o processo de simplificação. O objetivo é impedir a dizimação de vértices que possam causar erros no mapeamento da textura colorida durante a renderização. Dessa forma, obtém-se uma abordagem de compressão 3D progressiva orientada por dados (dados de mapeamento de textura). O método proposto tem como resultado: um modelo base; dados de refinamento que permitem a reconstrução do modelo de entrada; a textura de células a partir da qual são geradas as coordenadas de textura; e a textura colorida compartilhada por todos os modelos intermediários.

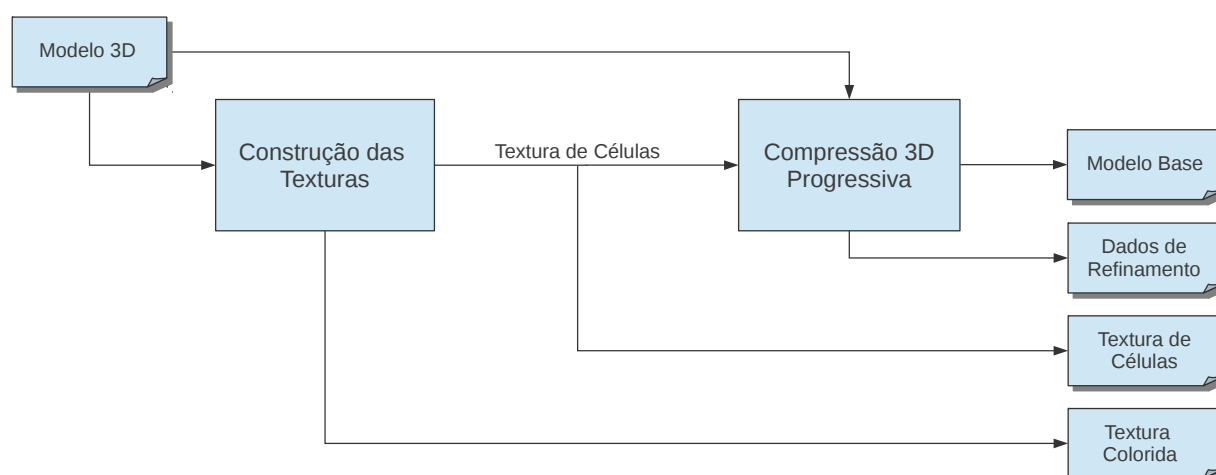


Figura 13 – Fluxograma da abordagem de compressão 3D progressiva orientada por dados proposta. A partir do modelo 3D de entrada são geradas a textura colorida e a textura de células. Esta textura é usada para guiar o processo de simplificação e assim garantir o correto mapeamento da textura colorida para todos os modelos intermediários. A compressão progressiva tem como resultado um modelo base, dados de refinamento, uma textura de células e uma textura colorida.

5.2 CONSTRUÇÃO DAS TEXTURAS

O método de mapeamento de textura por complexos de células cúbicas constrói as texturas utilizando dois modelos 3D de entrada: o modelo de alta resolução e sua versão simplificada. Na compressão 3D progressiva tem-se uma sequência de modelos intermediários que são versões simplificadas do modelo de entrada. Assim, apenas o modelo de alta resolução é utilizado para a construção das texturas.

Um complexo de células cúbicas consiste de um conjunto de células cúbicas representando o domínio de mapeamento da textura. Dado o modelo de entrada, o algoritmo gera uma grade de células cúbicas englobando toda a malha 3D. Cubos dessa grade que interceptam a malha formam o complexo de células cúbicas. A superfície no interior de cada célula cúbica é projetada nas faces dessa estrutura. O tipo

da projeção (paralela, perspectiva ou axial) associada a cada célula cúbica é feita de acordo com as suas faces de projeção. Detalhes de como é feita essa parametrização podem ser vistos em (APAZA-AGÜERO; SILVA; BELLON, 2011). A construção das texturas ocorre da seguinte maneira:

1. Define-se o tamanho da célula cúbica a partir das características principais da geometria do modelo de entrada.
2. Gera-se uma grade de cubos com o tamanho definido a partir do eixo alinhado (*axis-aligned bounding cube*).
3. Calcula-se quais cubos da grade são interceptados pela superfície da malha: cubos que são interceptados são chamados de células cúbicas.
4. Para cada célula cúbica, projeta-se a superfície contida em seu interior nas faces dessa estrutura (ver (APAZA-AGÜERO; SILVA; BELLON, 2011)).
5. A estrutura do complexo de células cúbicas, a qual contém as projeções associadas a cada célula cúbica, é armazenada na textura de células. As cores do modelo de alta resolução são armazenadas numa textura colorida do tipo azulejo (*tile texture*).

Por fim, a textura de células armazena: o tamanho da célula cúbica; o número de células cúbicas nos três eixos (x,y,z); e uma sequência de dados para cada cubo da grade. Os cubos que fazem parte do complexo de células cúbicas contêm o valor do centro de projeção seguido pelo identificador de cada face de projeção (6 faces). Esse identificador permite acessar o azulejo referente à face de projeção na textura colorida.

5.3 COMPRESSÃO 3D PROGRESSIVA

Como descrito no Capítulo 4, cada iteração do algoritmo de compressão realiza a dizimação de *patches* em duas etapas: conquista para dizimação e conquista para limpeza. A conquista para dizimação realiza uma travessia em largura baseada em *gates* de *patch* em *patch*. A partir de um *gate* de entrada, caso o vértice frontal possua valência entre 3 e 6, o *patch* é dizimado gerando uma nova configuração: o vértice frontal é removido e o *patch* é deterministicamente retriangulado. Caso

contrário, a face frontal é definida como *patch* nulo. De forma similar, a etapa de conquista para limpeza remove vértices com valência igual a 3 para malhas regulares. O algoritmo resulta em uma versão simplificada do modelo de entrada e dados da codificação da conectividade (sequência de códigos de valência ou *patch nulo*). O modelo de saída é usado como entrada da iteração seguinte, e o processo continua até obter o modelo base desejado.

5.3.1 Simplificação Orientada por Dados de Mapeamento de Textura

A simplificação orientada por dados de mapeamento de textura tem como objetivo prevenir a dizimação de *patches* cuja nova configuração possa resultar no mapeamento incorreto da textura colorida. Se a textura colorida for aplicada aos modelos intermediários sem a simplificação orientada por dados, esses modelos podem ser renderizados com artefatos (Figura 14).



Figura 14 – Visualização do modelo 3D Monkey: artefatos são renderizados se a textura colorida for aplicada aos modelos intermediários sem a simplificação orientada por dados de mapeamento de textura.

Durante a conquista para dizimação, aplica-se o Algoritmo 2 para cada *patch* a ser dizimado. Primeiro, calcula-se a nova configuração do *patch*, ou seja, a sua dizimação é simulada. Depois, gera-se uma *bounding box* de células cúbicas englobando todos os vértices da borda do *patch* (Figura 15). Verifica-se para cada face da nova configuração se ela intercepta algum cubo da *bounding box* que não faz parte do complexo de células cúbicas. Essa informação é obtida acessando a textura de células, a qual contém os dados de cada cubo da grade. Cubos que não fazem parte do complexo de células cúbicas não possuem dados de projeção associados. As-

sim, o *patch* pode ser dizimado apenas se todas as faces da nova configuração forem interceptadas por células cúbicas.

Como descrito no Capítulo 4, a restrição imposta à dizimação dos *patches* penaliza a taxa de compressão da conectividade, uma vez que um código de *patch* nulo é adicionado. No entanto, os resultados mostram que as penalidades ocorrem com pouca frequência, além dos bons resultados visuais obtidos.

Algoritmo 2 Pseudo-código do algoritmo para guiar o processo de simplificação.

Requer: *complexo_de_celulas_cubicas*

Requer: *patch*

```

1: nova_configuracao  $\leftarrow$  Calcule a nova configuração do patch
2: bounding_box  $\leftarrow$  Gere uma bounding box englobando todos os vértices da borda
   do patch
3: mapeamento_incorreto  $\leftarrow$  falso
4: para cada face f da nova_configuracao faça
5:   para cada cubo c da bounding_box faça
6:     se f intercepta c então
7:       se c não pertence ao complexo_de_celulas_cubicas então
8:         mapeamento_incorreto  $\leftarrow$  verdadeiro
9:         Interrompe loop
10:    fim se
11:  fim se
12: fim para
13: fim para
14: se mapeamento_incorreto == falso então
15:   Confirme a nova_configuracao
16: fim se

```

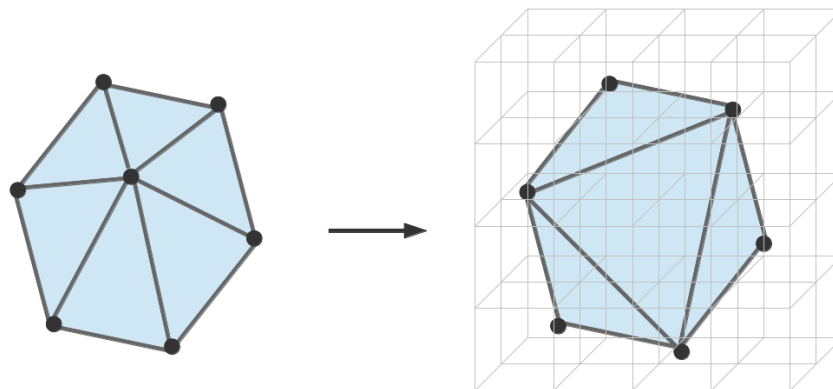


Figura 15 – Esquerda: *patch* com valência 6. Direita: *bounding box* englobando o *patch* com a nova configuração.

5.4 VISUALIZAÇÃO DOS MODELOS INTERMEDIÁRIOS

A aplicação da textura colorida no modelo 3D é realizada nos *shaders*¹ acessando a textura de células, e depende da forma como a estrutura de células cúbicas foi armazenada nessa textura. A parte mais relevante do algoritmo de aplicação ocorre no *fragment shader*: o fragmento é transformado a um ponto em coordenadas do mundo. Calcula-se então a qual cubo da grade este ponto é pertencente. Após a identificação do cubo, obtém-se o valor do centro de projeção da célula cúbica. O ponto é então projetado na face da célula cúbica usando o centro de projeção para obter o ponto de interseção. Esse ponto de interseção identifica a posição na textura de células que armazena os dados referentes à face de projeção (uma das 6 faces dependendo do tipo de projeção). A partir desses dados, calcula-se as coordenadas de textura as quais permitem acessar a cor na textura colorida. Por fim, o fragmento é colorido. Não há relação de dependência entre a aplicação da textura colorida e a decodificação progressiva (descrita na Seção 4.1.2).

5.5 EXPERIMENTOS E COMPARAÇÕES

Os experimentos nesta seção foram conduzidos para os modelos 3D Monkey, Anta e Joel, usando um computador com a seguinte configuração: Intel(R) Core(TM)2 Quad Processor Q8200 (2.33 GHz), 4GB de memória e NVIDIA GeForce GTX 560.

A Tabela 10(a) apresenta o número de vértices e faces dos modelos 3D originais. Dados sobre as texturas de células e texturas coloridas construídas para cada modelo 3D são apresentadas na Tabela 10(b). Os tamanhos das texturas estão relacionados ao tamanho das células, o qual depende das características geométricas: para objetos que possuem formas regulares, a superfície pode ser parametrizada utilizando células maiores; enquanto objetos mais detalhados, como a escultura do profeta Joel, requerem células menores. O tamanho da célula foi fixado em 6 para grandes modelos e em 30 para os modelos menores. As Figuras 16, 17 e 18 mostram o correto mapeamento da textura colorida, a qual é compartilhada por todos os modelos intermediários.

¹Um shader é um programa executado na GPU, usado para modificar o processo de renderização tradicional.

Tabela 10 – Informações sobre os modelos 3D testados: (a) número de vértices e faces do modelo 3D original; e (b) texturas de células e texturas coloridas.

(a)

Modelo 3D	#vértices	#faces
Monkey	50503	101002
Anta	206739	413470
Joel	1974038	3946493

(b)

Textura	Textura de Células				Textura Colorida		
	Tamanho Célula	Grade	Resolução	PNG	Resolução	PNG	JPEG
Monkey	30	22x35x24	256x128	19,5 KB	2560x2528	3,4 MB	1,2 MB
Anta	30	35x16x19	256x128	12,5 KB	2048x2016	2,1 MB	920,8 KB
Joel	6	51x63x171	1024x1024	188,9 KB	2120x2120	4,4 MB	2,7 MB

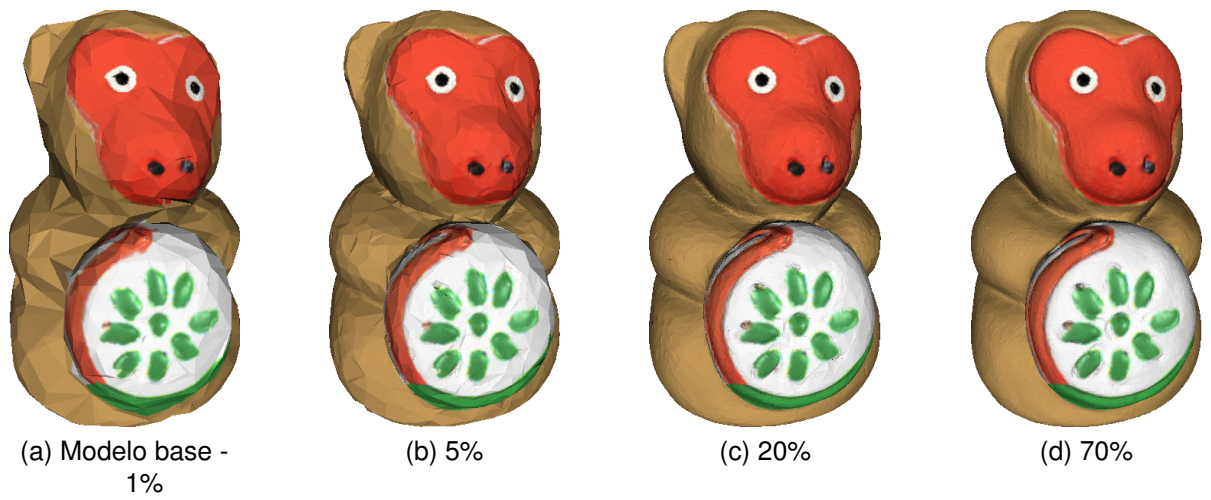


Figura 16 – Visualização do modelo 3D Monkey: modelos intermediários compartilhando a mesma textura colorida.

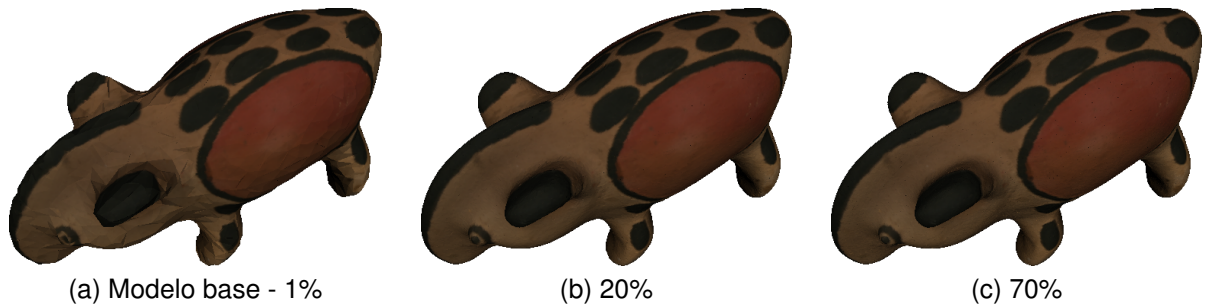


Figura 17 – Visualização do modelo 3D Anta: modelos intermediários compartilhando a mesma textura colorida.



Figura 18 – Visualização do modelo 3D Joel: modelos intermediários compartilhando a mesma textura colorida.

Nesta seção, a abordagem proposta é denominada AD + CCC, abreviação para a combinação da compressão 3D progressiva baseada no método de Alliez e Desbrun com o mapeamento de texturas por complexo de células cúbicas. AD + CCC foi comparada a outras duas abordagens baseadas no método de Alliez e Desbrun: Lee, Lavoué e Dupont (2010) para codificação do atributo de cor (LLD); e uma versão desta adicionando a métrica para a preservação das cores proposta em seu trabalho posterior (LEE; LAVOUÉ; DUPONT, 2012) (LLD + métrica cor). Essas duas abordagens são descritas em detalhes no Capítulo 4. As três abordagens testadas são derivadas de uma única implementação do algoritmo de Alliez e Desbrun, mantendo iguais a precisão da quantização dos vértices. As diferenças entre as abordagens são: LLD não possui restrições para a dizimação dos vértices; LLD + métrica cor impede a dizimação de vértices que possam causar distorção visual baseado no atributo de cor; enquanto AD + CCC impede a dizimação de vértices que possam causar erros no mapeamento da textura colorida.

5.5.1 Taxa de Compressão

A Tabela 11 compara as taxas de *bits* requeridas para a reconstrução da conectividade do modelo 3D original (A) a partir do modelo base (B) para as três abordagens de compressão. A capacidade de compressão da conectividade para ambas as abordagens orientadas por dados está diretamente relacionada às restrições à dizimação dos vértices impostas por cada uma. Ao preservar as cores, LLD + métrica cor

causa muitas penalidades. O custo adicional requerido para o armazenamento dos dados de conectividade é significativo para grandes modelos, como o modelo 3D Joel. Apesar de também ser orientada por dados, a restrição imposta por AD + CCC causa poucas penalidades, apresentando taxas de compressão da conectividade similar à abordagem que não é orientada por dados.

Tabela 11 – Taxas de compressão dos modelos testados em *bits* por vértice. Quantização dos vértices usando 10 *bits*.

(a) Monkey			
Abordagem	#vértices $A \rightarrow B$	Iterações	Conectividade (bit/v)
LLD	50503 \rightarrow 841	11	8,747
LLD + métrica cor	50503 \rightarrow 831	15	11,126
AD + CCC	50503 \rightarrow 835	12	8,909

(b) Anta			
Abordagem	#vértices $A \rightarrow B$	Iterações	Conectividade (bit/v)
LLD	206739 \rightarrow 2163	12	8,282
LLD + métrica cor	206739 \rightarrow 2182	26	13,841
AD + CCC	206739 \rightarrow 2183	12	8,285

(c) Joel			
Abordagem	#vértices $A \rightarrow B$	Iterações	Conectividade (bit/v)
LLD	1974038 \rightarrow 22844	12	8,390
LLD + métrica cor	1974038 \rightarrow 23304	32	18,760
AD + CCC	1974038 \rightarrow 23843	12	8,410

Como observado nos experimentos do Capítulo 4, a restrição imposta pela métrica cor pode gerar superfícies com regiões mais simplificadas do que outras. Com tal característica, LLD + métrica cor possui como desvantagem a possível deformação da topologia a baixas resoluções. A Figura 19 mostra a renderização em *wireframe* dos modelos base Monkey obtidos com as três abordagens de compressão. O modelo base AD + CCC apresenta a superfície suave, assim como o modelo base LLD.

5.5.2 Qualidade Visual

Para a avaliação da qualidade visual foram selecionadas as vistas dos objetos que apresentam detalhes visuais significativos e coloridos: a região das costas da Anta destacando a pintura indígena; o pergaminho do profeta Joel com as passagens bíblicas; e a região frontal do Monkey.

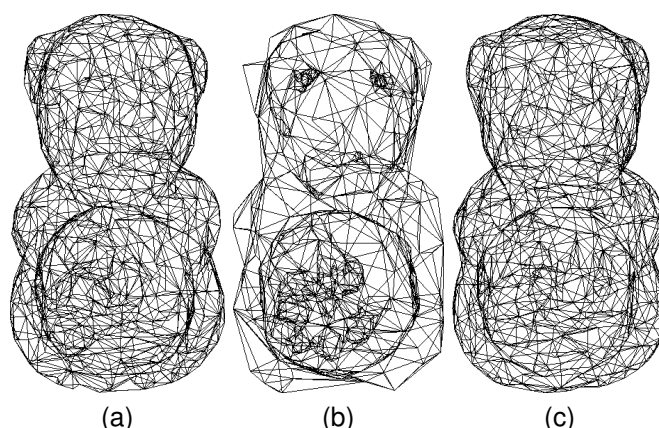


Figura 19 – Diferenças visuais entre os modelos base Monkey: (a) LLD, (b) LLD + métrica cor e (c) AD + CCC.

As Tabelas 12, 13 e 14 mostram as curvas de distorção visual e os resultados visuais obtidos com as abordagens de compressão. As curvas ilustram os valores RMSE em função da porcentagem de elementos. As figuras das tabelas mostram os resultados visuais de modelos intermediários em baixa resolução com similar número de elementos (marcados nas curvas com \square e \times). Calculou-se o RMSE vista a vista do modelo intermediário com o modelo 3D original. A vista selecionada para o cálculo do RMSE é apresentada ao lado do gráfico. Os modelos foram renderizados utilizando um *fragment shader*. Os fragmentos são coloridos sem iluminação e sem sobre-amento para minimizar a influência das diferenças geométricas. Para AD + CCC, é aplicada a textura colorida em formato PNG (formato que suporta a compressão de imagens sem perdas de dados).

De acordo com as curvas de distorção, LLD + métrica cor inicia com maior distorção visual porém logo supera LLD ao preservar as cores. LLD + métrica cor preserva as regiões delimitadas por descontinuidade de cor, como nos detalhes frontais do Monkey e nas costas da Anta. As curvas AD + CCC iniciam com menor distorção visual em comparação às abordagens que usam atributo de cor. Os modelos intermediários AD + CCC apresentam boa aproximação ao modelo original em relação às cores.

O mapeamento de texturas possui como vantagem a capacidade de adição de detalhes à superfície quando há restrição geométrica para tal, efeito conhecido como mascaramento visual. Esse efeito é visualmente significativo a baixas resoluções, principalmente para grandes modelos, como o modelo 3D Joel. Como resultado, AD + CCC provê a transição suave entre os modelos intermediários.

Tabela 12 – Avaliação da qualidade visual do modelo 3D Monkey para as abordagens de compressão. Curvas de distorção visual obtidas através do RMSE vista a vista. Comparação dos modelos intermediários com similar porcentagem de elementos.

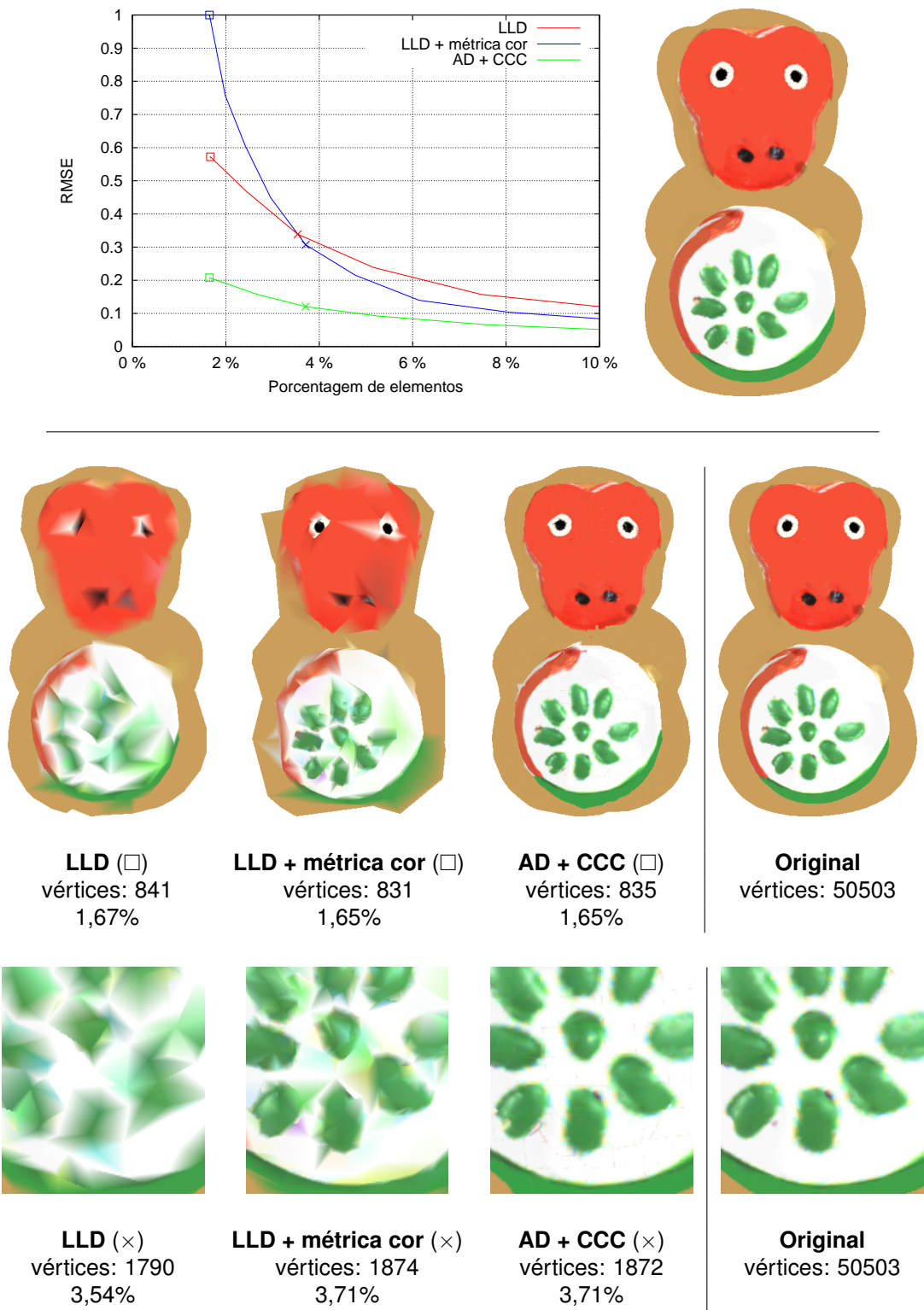


Tabela 13 – Avaliação da qualidade visual do modelo 3D Anta para as abordagens de compressão. Curvas de distorção visual obtidas através do RMSE vista a vista. Comparação dos modelos intermediários com similar porcentagem de elementos.

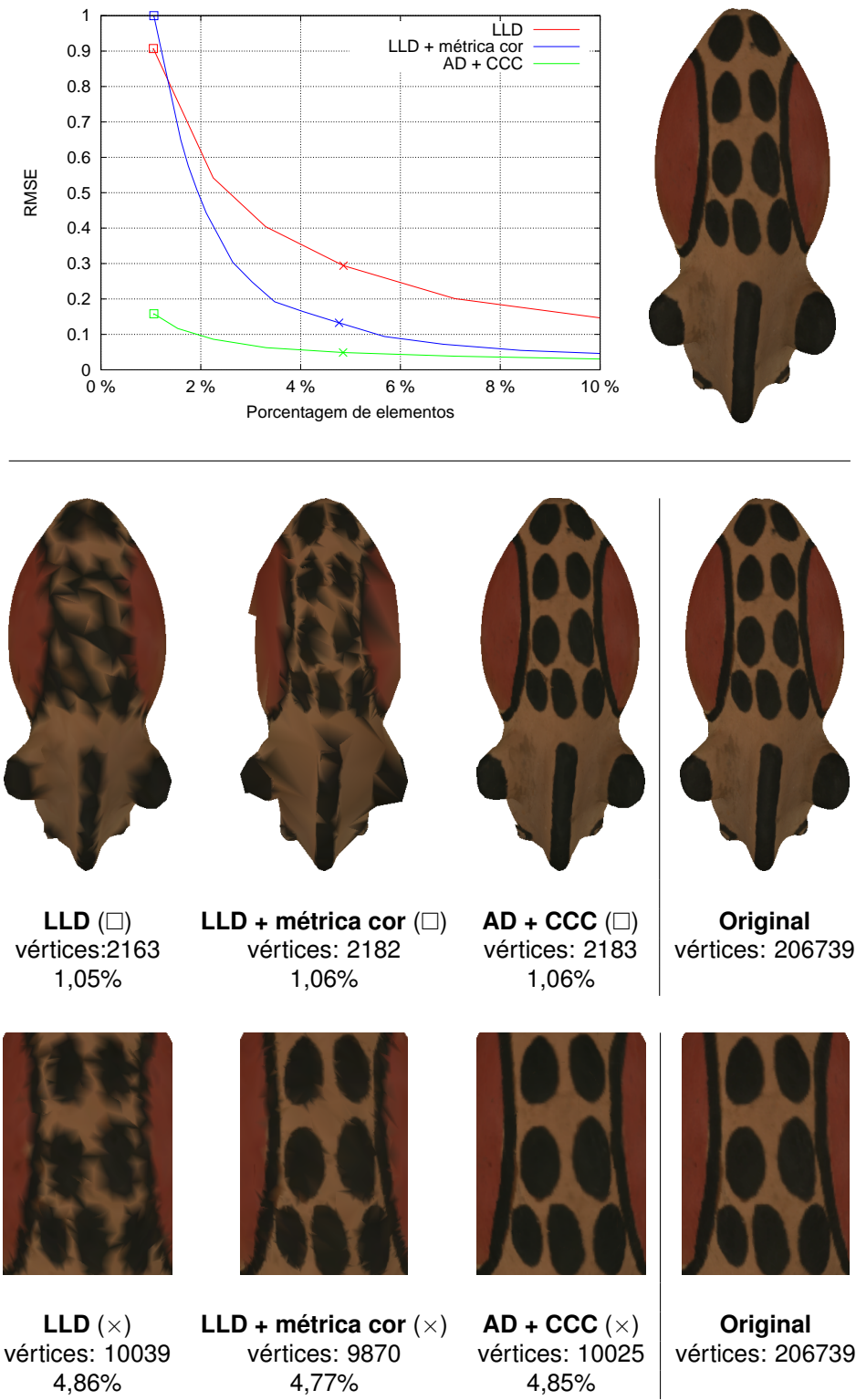


Tabela 14 – Avaliação da qualidade visual do modelo 3D Joel para as abordagens de compressão. Curvas de distorção visual obtidas através do RMSE vista a vista. Comparação dos modelos intermediários com similar porcentagem de elementos.

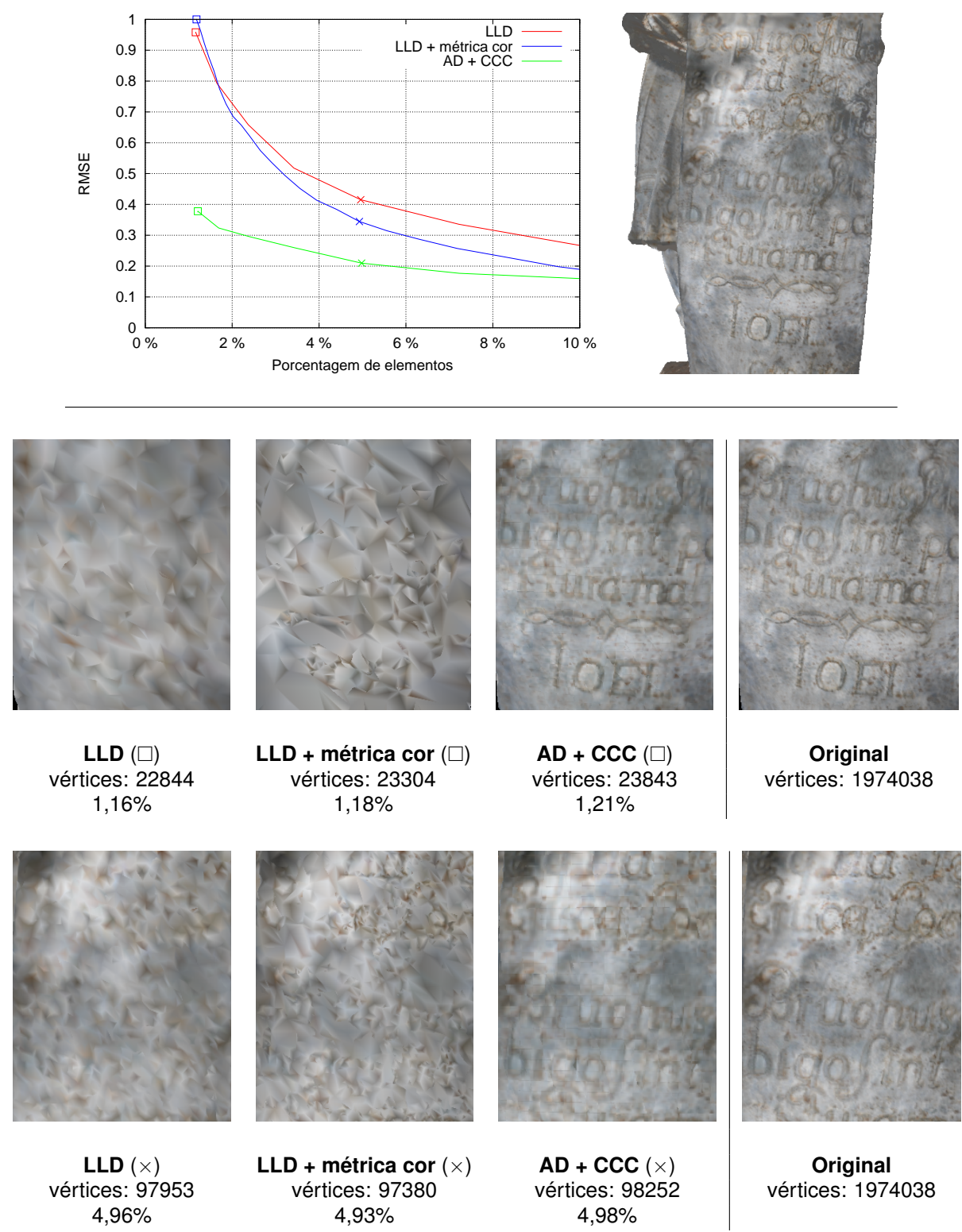


Tabela 15 – Custo para representar as coordenadas de textura: solução tradicional vs AD + CCC.

Modelo 3D	#vértices	$C_{tradicional}$	C_{AD+CCC}
Monkey	50503	394,55 KB	19,5 KB
Anta	206739	1,58 MB	12,5 KB
Joel	1974038	15,06 MB	188,9 KB

5.5.3 Custo de Armazenamento

Em geral, modelos 3D com textura requerem maior espaço de armazenamento que modelos 3D com atributo de cor, pois algumas texturas podem ser maiores que o próprio modelo 3D. Para modelos 3D com atributo de cor e pouco coloridos, as cores podem ser bem compactadas com técnicas de entropia ou tabelas de mapeamento (LEE; LAVOUÉ; DUPONT, 2010). O mapeamento de texturas para modelos 3D progressivos requer o armazenamento de uma ou mais texturas coloridas para os modelos intermediários. No entanto, o mapeamento de texturas possibilita a adição de detalhes às superfícies e aumenta o realismo dos modelos 3D.

A abordagem AD + CCC é apropriada à compressão, uma vez que usa uma estrutura de dados compacta para representar as coordenadas de textura (textura de células). Uma única textura colorida é compartilhada por todos os modelos intermediários. Além disso, a textura colorida pode ser comprimida, por exemplo, no formato JPEG.

Considere um modelo 3D com N vértices mapeado por uma textura. Na solução tradicional, as coordenadas de texturas (v, t) são representadas em ponto flutuante (4 bytes float cada uma). Assim, o custo para representar as coordenadas de textura é $C_{tradicional} = N * (2 * 4)$. Já o custo da abordagem proposta é $C_{AD+CCC} = T_{células}$, ou seja, o tamanho da textura de células (formato PNG). A Tabela 15 apresenta o custo da solução tradicional e da abordagem proposta. Para os modelos 3D testados, o custo de AD + CCC é menor, e a diferença é mais significativa para grandes modelos (e.g. modelo 3D Joel).

5.5.4 Carregamento Progressivo da Textura Colorida

A compressão progressiva possibilita a visualização dos modelos intermediários na reconstrução 3D progressiva. No entanto, a textura colorida precisa ser carregada antes de iniciar a visualização. Assim, a latência de visualização é determinada

pelo tamanho dessa textura, a qual pode ser maior em tamanho em comparação ao modelo 3D comprimido. Uma maneira de diminuir a latência é através do carregamento progressivo da textura colorida. O algoritmo JPEG, frequentemente utilizado para comprimir imagens de textura, suporta a codificação progressiva. Nesse modo, as componentes de cor da imagem são codificadas em múltiplas varreduras (*scans*). O primeiro *scan* produz uma versão em baixa resolução, enquanto os *scans* seguintes aumentam gradualmente a resolução da imagem.

O gráfico da Figura 20 ilustra a porcentagem de elementos decodificados em função do tempo na reconstrução progressiva do modelo 3D Joel, o qual possui o maior arquivo de textura colorida (2,7 MB). O gráfico mostra a latência usando a textura colorida comprimida em JPEG progressivo e JPEG sequencial. Para uma velocidade de conexão com a Internet de 10 Mbit/s, o primeiro *scan* da textura colorida é realizado em menos de 1s. O JPEG progressivo possibilita iniciar a visualização aproximadamente 2s antes do JPEG sequencial, porém requer aproximadamente 3s a mais para finalizar a decodificação da imagem. Apesar de diminuir a latência, o carregamento progressivo da textura colorida é vantajoso apenas se a decodificação total da imagem for mais rápida que a velocidade da transferência de dados.

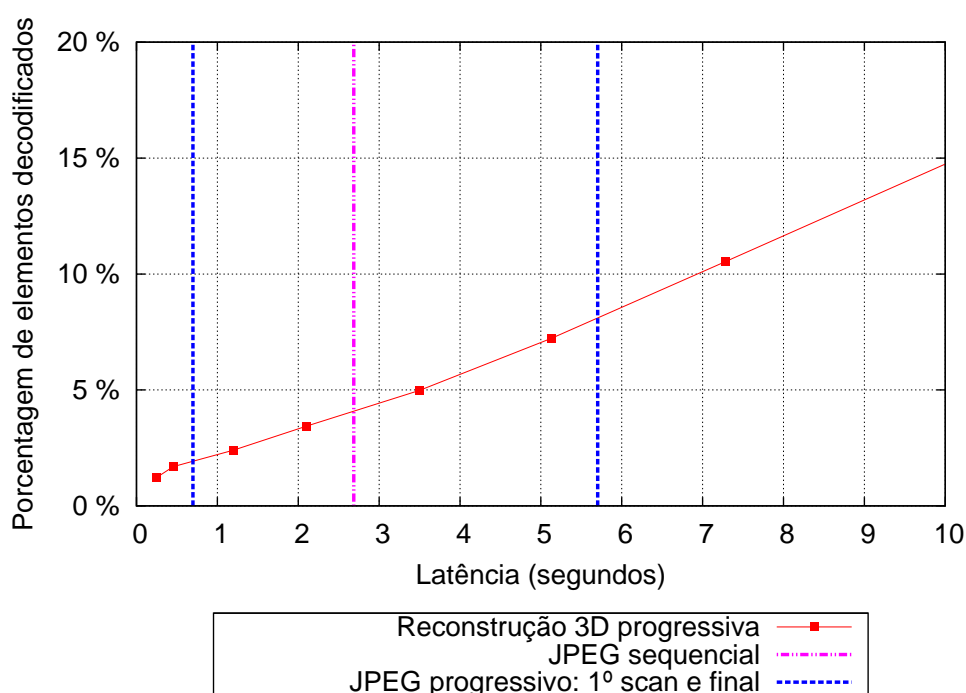


Figura 20 – Latência de visualização na reconstrução progressiva do modelo 3D Joel (vermelho) usando a textura colorida comprimida em JPEG progressivo (azul) e JPEG sequencial (rosa).

A Figura 21 mostra a renderização do modelo 3D Joel após o primeiro e segundo *scans*. A textura colorida deixa de apresentar o aspecto “quadriculado” após o segundo *scan*. Este experimento foi realizado usando um visualizador 3D e decodificador JPEG implementado em C++, OpenGL e a biblioteca libjpeg².

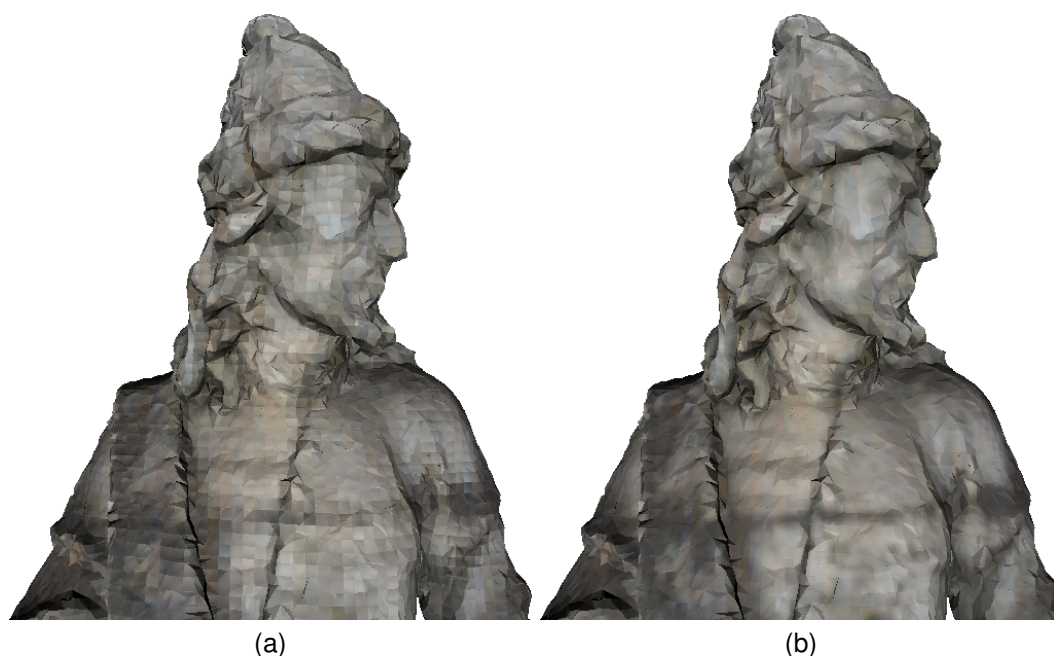


Figura 21 – Renderização do modelo 3D Joel com a textura comprimida com JPEG progressivo: (a) primeiro *scan* e (b) segundo *scan*.

5.6 CONCLUSÃO

Este capítulo apresentou uma nova abordagem de compressão 3D progressiva orientada por dados com o objetivo de melhorar a qualidade visual de modelos 3D coloridos. As cores são representadas por uma textura colorida ao invés de atributo de cor. São construídas duas texturas: a textura de células contendo o complexo de células cúbicas, com a qual são geradas as coordenadas de textura durante a visualização baseada em GPU; e uma textura colorida contendo as cores do modelo 3D original, a qual é compartilhada por todos os modelos intermediários. Na compressão 3D progressiva, os dados do mapeamento de textura contidos na textura de células são usados para guiar o processo de simplificação impedindo a dizimação de vértices que possam causar erros no mapeamento da textura colorida.

²libjpeg.sourceforge.net

Os resultados dos experimentos mostram que a abordagem proposta é apropriada à compressão progressiva de modelos 3D coloridos. A restrição imposta à dizimação dos vértices para a preservação do mapeamento da textura colorida gera poucas penalidades, pouco interferindo na capacidade de compressão da conectividade. As coordenadas de texturas são representadas por meio de uma estrutura de dados compacta armazenada em uma pequena textura, enquanto a textura colorida pode ser bem compactada usando algoritmos de compressão de imagens. A textura colorida é corretamente compartilhada por todos os modelos intermediários, aumentando significativamente a qualidade visual a baixas resoluções.

Prover boa qualidade visual a baixas resoluções é essencial para aplicações executadas em ambientes com recursos limitados ou que precisam restringir o uso de dados de geometria, como na visualização de modelos 3D realistas. Através do mapeamento de textura, a abordagem proposta provê a transição suave entre os modelos intermediários, os quais apresentam boa aproximação ao modelo original em relação às cores. Assim, a abordagem proposta mostra-se uma solução eficiente para a reconstrução 3D progressiva de modelos 3D onde a qualidade visual é requisito primário. Além disso, a ideia principal de representar as cores do modelo 3D através do mapeamento de texturas por complexo de células cúbicas pode ser adaptada a outros métodos progressivos, como *Progressive Meshes* ou abordagens multi-resolução *view-dependent*.

6 FRAMEWORK DE VISUALIZAÇÃO 3D BASEADA NA WEB: UM CASO DE ESTUDO NA VISUALIZAÇÃO INTERATIVA DE MODELOS 3D DE ESCULTURAS DE ALEIJADINHO

O Grupo IMAGO de Pesquisa da Universidade Federal do Paraná, em parceria com o IPHAN e a UNESCO, realizou a digitalização 3D de esculturas barrocas feitas por Aleijadinho. Este capítulo apresenta um caso de estudo na visualização remota de modelos 3D do conjunto de esculturas chamado Os Doze Profetas, uma das obras mais conhecidas do artista. O *framework* de visualização 3D baseada na *web* apresentado neste capítulo complementa o processo de preservação digital e divulgação da obra de Aleijadinho.

As esculturas dos profetas de Aleijadinho estão localizadas no Santuário de Bom Jesus de Matosinhos em Congonhas, Minas Gerais, considerado patrimônio mundial pela UNESCO. O Santuário foi construído entre os séculos XVIII e XIX, sendo composto por uma igreja, um adro dos profetas e seis capelas. O adro contém doze esculturas de profetas do velho testamento, com tamanho próximo ao natural, dispostos em ordem simétrica e alternada de acordo com a citação na Bíblia. As esculturas foram feitas em pedra sabão, um material macio e fácil de ser danificado. Localizadas em ambiente externo com visitação livre, as esculturas sofrem a infestação de líquens, ações de vandalismo, além da própria degradação natural.

O processo de digitalização 3D desenvolvido pelo grupo IMAGO tem como objetivo a geração de modelos 3D com alto nível de fidelidade aos objetos físicos. O realismo dos modelos 3D é alcançado devido à precisão geométrica e à preservação da aparência do objeto. Assim, os modelos 3D são comprimidos a fim de facilitar a sua transmissão e diminuir a latência de visualização. Para tal, foi utilizada a abordagem de compressão 3D progressiva orientada por dados de mapeamento de textura descrita no Capítulo 5. Os dados obtidos nesse processo são armazenados em uma base de dados de apoio à reconstrução 3D progressiva. Por fim, os modelos 3D são disponibilizados para visualização através de uma aplicação *web*.

Além do grupo IMAGO, a Universidade de São Paulo (Museu de Ciências e Pró-reitoria de Cultura e Extensão) também realizou a digitalização 3D das esculturas dos profetas de Aleijadinho (RODRIGUES *et al.*, 2013). Recentemente, Lavoué, Chevalier e Dupont (2013) propuseram uma solução técnica para a visualização remota de modelos 3D na *web*. A abordagem proposta provê a transmissão e a visualização

de modelos 3D compridos através do navegador *web*. As principais diferenças destes trabalhos em relação ao *framework* proposto são apresentadas na Seção 6.7.

6.1 DIGITALIZAÇÃO 3D

Para a digitalização 3D das esculturas foi utilizado um *pipeline* composto por três etapas principais: aquisição dos dados, reconstrução 3D e geração de textura. As imagens de profundidade e imagens coloridas foram obtidas utilizando um *scanner a laser* (Minolta Vivid 910) e uma câmera digital colorida (Canon EOS5D). A reconstrução 3D é composta por: (1) registro ou alinhamento; (2) integração; (3) preenchimento de buracos; e (4) geração da malha. Por fim, é gerada uma textura de alta qualidade a partir das fotografias de alta resolução capturadas na aquisição. Este processo consiste em uma combinação de algoritmos desenvolvidos por alunos de mestrado e doutorado do grupo IMAGO. Os resultados da preservação digital 3D da escultura do profeta Joel foram publicados no *Journal of Cultural Heritage* (ANDRADE *et al.*, 2011).

6.2 COMPRESSÃO 3D PROGRESSIVA ORIENTADA POR DADOS

Como descrito no Capítulo 5, a abordagem de compressão 3D progressiva orientada por dados proposta nesta tese provê a melhoria da qualidade visual de modelos 3D coloridos por meio do mapeamento de textura. Baseado no mapeamento por complexo de células cúbicas, são construídas duas texturas: a textura de células contendo o complexo de células cúbicas, com a qual são geradas as coordenadas de textura durante a visualização baseada em GPU; e uma textura colorida contendo as cores do modelo 3D original, a qual é compartilhada por todos os modelos intermediários. Na compressão 3D progressiva, os dados do mapeamento de textura contidos na textura de células são usados para guiar o processo de simplificação impedindo a dizimação de vértices que possam causar erros no mapeamento da textura colorida.

6.3 BASE DE DADOS

O diagrama da Figura 22 provê uma visão geral da base de dados usada para apoiar a reconstrução 3D progressiva. A Tabela *tb_modelo* armazena informações sobre cada modelo 3D gerado no processo de digitalização 3D. A Tabela *tb_mapeamento_ccc*

relaciona o resultado da construção das texturas: a textura de células (*tb_textura_celulas*) e a textura colorida (*tb_textura_colorida*). Na Tabela *tb_compressao* consta o mapeamento por complexo de células cúbicas utilizado na compressão 3D progressiva orientada por dados. Os dados de refinamento gerados a cada iteração do algoritmo de compressão são armazenados na Tabela (*tb_iteracao*). Por fim, as informações sobre o modelo base obtido na última iteração são armazenadas na Tabela *tb_modelo_base*.

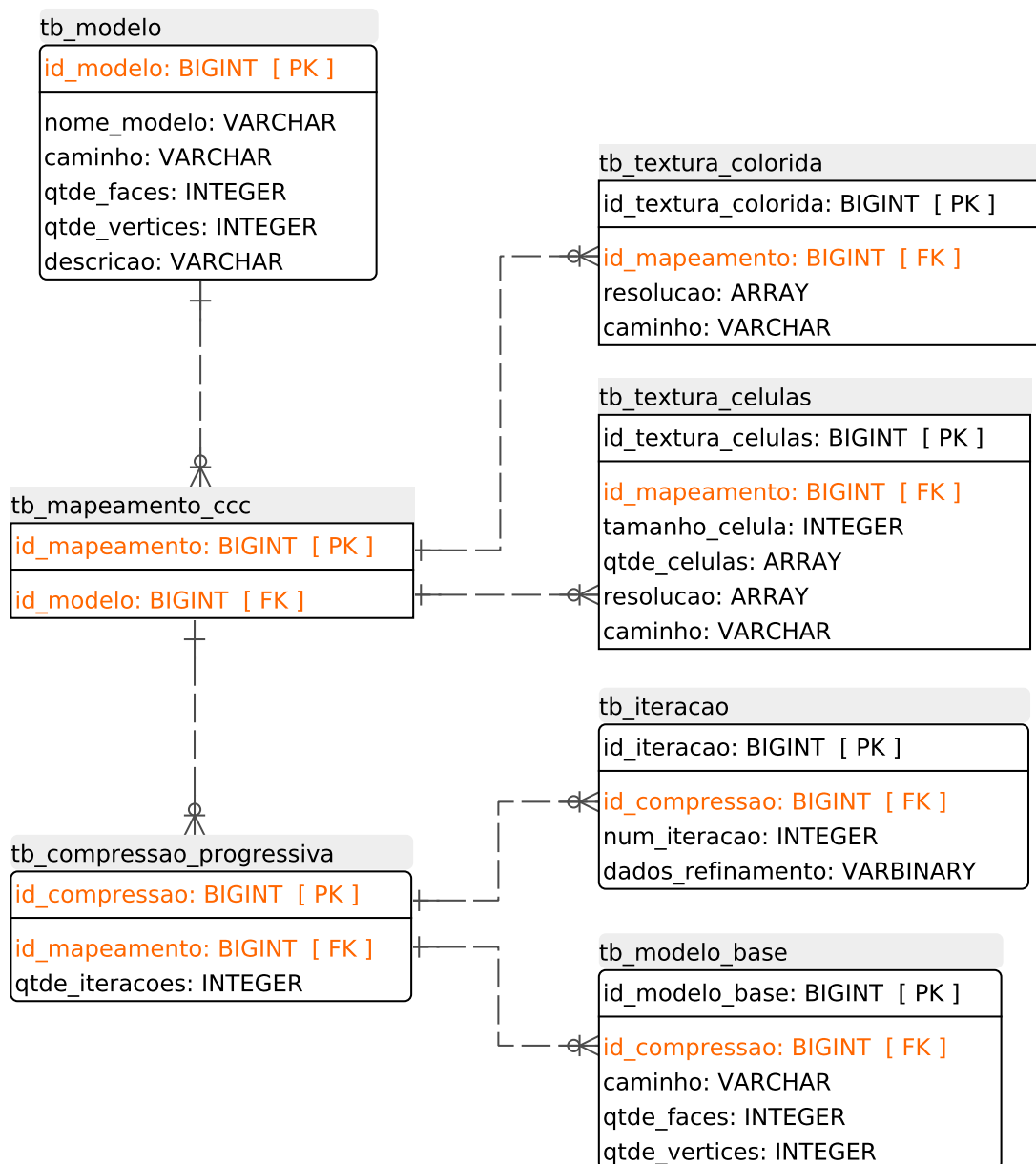


Figura 22 – Diagrama do modelo de dados usado para apoiar a reconstrução 3D progressiva.

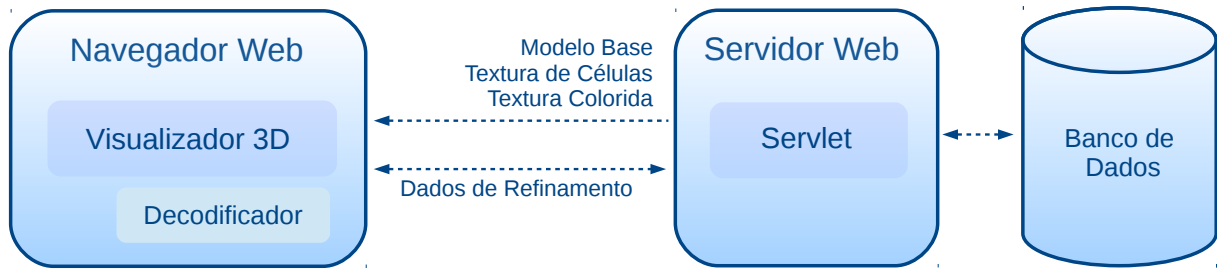


Figura 23 – Diagrama da aplicação web.

6.4 APLICAÇÃO WEB

A visualização remota dos modelos 3D dos profetas é feita através de uma aplicação web, composta por um visualizador 3D e um servidor web conectado a um banco de dados. A Figura 23 mostra o diagrama da aplicação. O visualizador 3D recebe como entrada um modelo base, uma textura de células e uma textura colorida, e os dados de refinamento são solicitados ao servidor. Devido ao mapeamento de textura, a visualização do modelo 3D inicia após o carregamento das texturas na memória da placa de vídeo. O usuário pode interagir livremente com o modelo 3D através do *mouse* ou toque na tela (*touch screen*), e ainda há a opção da interface gráfica e teclas de atalho (Figura 24).

O visualizador 3D consiste em uma página web desenvolvida em HTML5, JavaScript e WebGL. A WebGL possibilita a exibição de gráficos 3D no navegador web sem a instalação de *plugins*. Com a linguagem JavaScript é possível criar interfaces dinâmicas. No entanto, esta linguagem possui uma limitação que é sua natureza *single-threaded*, ou seja, o código JavaScript só pode ser executado em uma única *thread*. Felizmente, o HTML5 através dos *Web Workers* provê uma maneira de executar *scripts* em *background*. Assim, a reconstrução progressiva do modelo 3D pode ser realizada em paralelo à interação do usuário.

6.4.1 Reconstrução 3D Progressiva

Seja d_0, d_1, \dots, d_{I-1} a sequência de dados de refinamento obtidos com I iterações do algoritmo de compressão 3D progressiva. A reconstrução 3D progressiva consiste em um processo iterativo de refinamento do modelo 3D, que tem como resultado a sequência m_1, m_2, \dots, m_I de modelos intermediários, onde m_I é o modelo 3D original. Cada modelo intermediário é dado por:

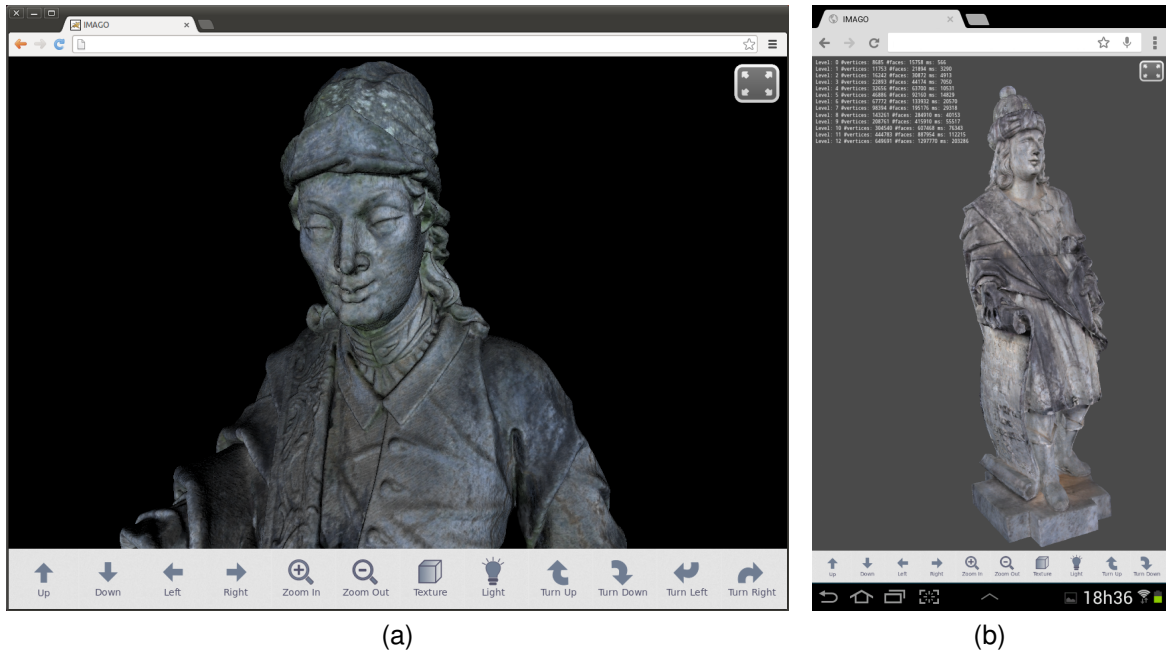


Figura 24 – Visualização de modelos 3D utilizando a aplicação *web* desenvolvida: (a) profeta Amós a partir de um computador e (b) profeta Baruc a partir de um *tablet*.

$$m_i = \text{decodificador}(m_{i-1}, d_{i-1}) \text{ para } i = \{1, 2, \dots, I\} \text{ e } m_0 = \text{modelo base} \quad (7)$$

O decodificador recebe como entrada um modelo intermediário e os dados de refinamento correspondentes a este modelo, realiza a decodificação (algoritmo descrito na Seção 4.1.2) e por fim retorna o modelo 3D refinado. Este modelo é atualizado no *buffer* de renderização e depois retorna ao decodificador. Como descrito no Capítulo 5, o mapeamento da textura colorida ocorre a cada *frame* nos *shaders*, independentemente a este processo.

A sequência de dados de refinamento é solicitada ao servidor em paralelo antes de iniciar a reconstrução 3D progressiva. A decodificação de um modelo intermediário m_i é feita assim que m_{i-1} e d_{i-1} estão disponíveis. Uma pequena *servlet*¹ é responsável por atender às solicitações. Os dados de refinamento de cada modelo intermediário são acessados no banco de dados através do identificador da compressão progressiva e do número da iteração (*id_compressao* e *num_iteracao*). Essa abordagem de implementação permite controlar os níveis de reconstrução.

¹Programa Java que estende a funcionalidade do servidor *web*.

O algoritmo de decodificação requer memória adicional para acomodar os novos elementos. Uma grande quantidade de objetos precisa ser alocada e desalocada a cada iteração. No entanto, a linguagem JavaScript não possibilita o gerenciamento explícito da memória. Por isso, a maioria dos objetos é reutilizada a cada iteração. Este esforço de implementação visa minimizar as pausas causadas pela execução do *garbage collector*, as quais diminuem a taxa de *frames* por segundo (fps).

6.5 RESULTADOS EXPERIMENTAIS

Os experimentos nesta seção foram conduzidos para os modelos 3D dos profetas Joel, Baruc e Habacuc. Os resultados da compressão 3D progressiva orientada por dados são apresentados na Tabela 16. A taxa de *bits* requerida para reconstrução total da conectividade e da geometria do modelo 3D original (A) é dada em *bits* por vértice. Os modelos base (B) possuem em torno de 1,16% de elementos do modelo 3D original e foram obtidos com 12 iterações do algoritmo de compressão.

Tabela 16 – Resultado da compressão 3D progressiva orientada por dados: (a) textura de células e textura colorida; e (b) taxas de compressão dos modelos 3D testados. Quantização dos vértices usando 10 *bits*.

(a)						
Textura	Textura de Células				Textura Colorida	
	Tamanho Célula	Grade	Resolução	PNG	Resolução	JPEG
Joel	6	51x63x171	1024x1024	188,9 KB	2120x2120	2,7 MB
Baruc	6	57x52x171	1024x1024	195 KB	2128x2120	2,6 MB
Habacuc	6	72x55x171	1024x1024	172,6 KB	2054x2056	2,9 MB

(b)			
Modelo 3D	#vértices A	#vértices B (tamanho)	Conectividade + Geometria (bit/v)
Joel	1974038	23843 (336 KB)	15,98
Baruc	2887390	33463 (472 KB)	14,98
Habacuc	2924749	33808 (480 KB)	14,85

Para o experimento de visualização 3D remota foram utilizados um notebook Intel(R) Core(TM) i3 (2.30 GHz) com 4GB de memória, navegador *web* Google Chrome e velocidade de conexão com a Internet de 10 Mbit/s.

A Tabela 17 apresenta a comparação entre a compressão 3D progressiva e a transmissão do modelo 3D original no formato ASCII sem compressão e comprimido com ZIP. O tamanho informado para a compressão 3D progressiva é dado pela soma dos tamanhos do modelo base, textura de células e textura colorida. A compressão

do modelo 3D original com ZIP diminui a latência de visualização. A compressão progressiva requer um tempo maior para disponibilizar o modelo 3D original, porém o usuário espera apenas em torno de 3s para iniciar a visualização. Os gráficos da Figura 25 ilustram a porcentagem de elementos decodificados em função do tempo para os modelos 3D Joel e Habacuc. O tempo de reconstrução possui comportamento linear em relação à porcentagem de elementos. O tempo de reconstrução total é considerado rápido para uma implementação JavaScript.

Tabela 17 – Latência de visualização: transmissão do modelo 3D original no formato ASCII sem e com compressão (ZIP) e compressão 3D progressiva.

Modelo 3D	Sem compressão		Com compressão (ZIP)		Compressão 3D progressiva		
	Tamanho	Latência	Tamanho	Latência	Tamanho	Latência	T. Recons.
Joel	197 MB	2,60min	34 MB	26,52s	3,1 MB	2,97s	1,36min
Baruc	290 MB	3,88min	47 MB	37,16s	3,2 MB	3,17s	1,96min
Habacuc	295 MB	3,93min	49 MB	38,99s	3,4 MB	3,42s	2,02min

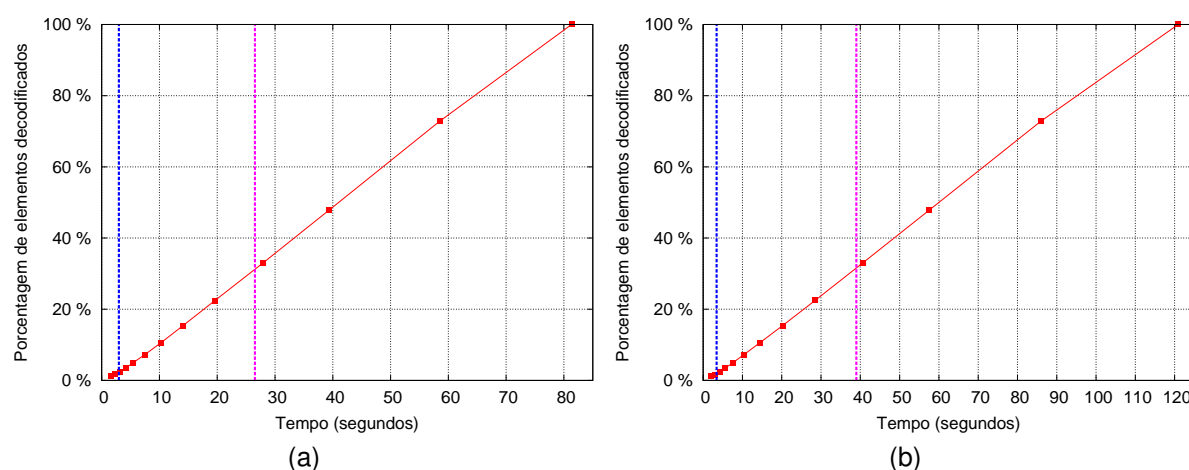


Figura 25 – Visualização remota dos modelos 3D: (a) Joel e (b) Habacuc. Reconstrução 3D progressiva em vermelho e latência em azul. Latência considerando a transmissão do modelo 3D original comprimido com ZIP em rosa.

As Figuras 26 e 27 apresentam os resultados visuais da reconstrução 3D progressiva. A abordagem de compressão 3D progressiva orientada por dados proposta nesta tese provê modelos intermediários com boa qualidade visual mesmo a baixas resoluções. A preservação das cores através do mapeamento de textura possibilita a transição suave entre os modelos intermediários. O primeiro modelo intermediário exibido para o usuário (em torno de 1%) apresenta aproximação razoável ao modelo 3D original (100%). Modelos intermediários em torno de 20% de elementos são visualmente similares ao modelo 3D original. Os modelos 3D são visualizados com aproximadamente 25-30 fps após a reconstrução total.

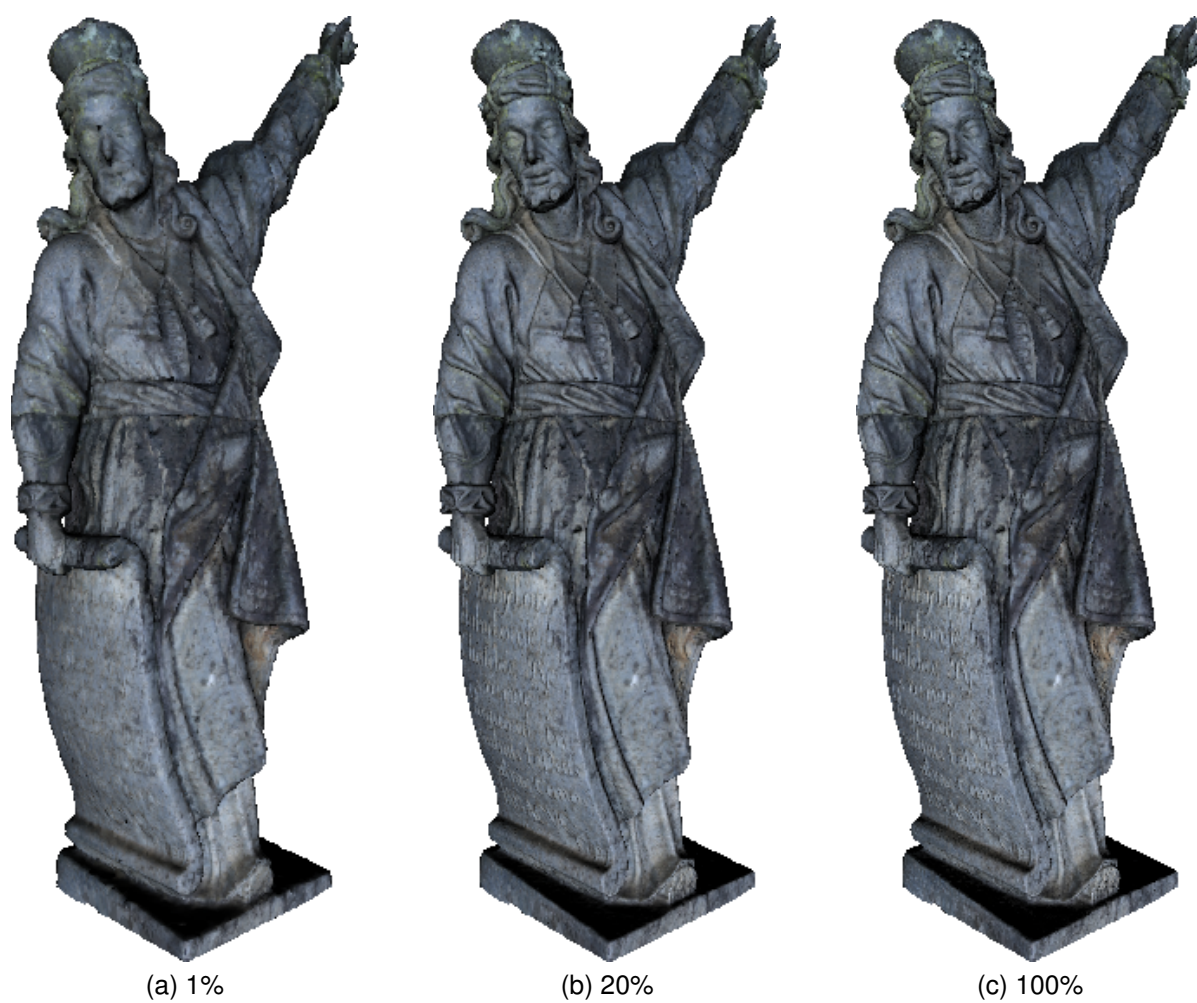


Figura 26 – Visualização de modelos intermediários na reconstrução progressiva do modelo 3D Habbuc.



Figura 27 – Visualização de modelos intermediários na reconstrução progressiva do modelo 3D Baruc.

O desempenho da reconstrução 3D progressiva depende do tempo de resposta da aplicação. O tempo de reconstrução apresentado na Tabela 17 considera o tempo de resposta quando o servidor atende um usuário por vez. No cenário em que há vários acessos simultâneos, o servidor tende a demorar para atender as solicitações dos usuários, aumentando assim o tempo de reconstrução. Para avaliar o desempenho da reconstrução 3D progressiva, a aplicação *web* foi submetida a uma bateria de testes simulando acessos simultâneos. Para tal, foi utilizada a ferramenta *open source* JMeter², a qual possibilita a realização de testes de desempenho, de carga e de *stress* em aplicações *web*. O Apache Tomcat³ atua como o servidor *web*, integrado com o sistema gerenciador de banco de dados PostgreSQL⁴.

Para a reconstrução total de cada modelo 3D, 12 requisições HTTP são solicitadas sequencialmente ao servidor (12 iterações \Rightarrow sequência de 12 dados de refinamento). A decodificação de um modelo intermediário pode ser realizada somente quando os dados de refinamento correspondentes estão disponíveis. Considerando essas definições, o tempo de reconstrução médio foi calculado simulando 50 e 100 acessos simultâneos para a visualização remota do modelo 3D Habacuc, o maior entre os modelos testados. Primeiro foi calculado o tempo de reconstrução médio obtido com 10 visualizações considerando um único acesso por vez. Depois, foi simulado o acesso simultâneo realizado por 50 e 100 usuários. Através do JMeter, obteve-se o tempo de resposta médio de cada uma das 12 requisições nas baterias de testes.

O gráfico da Figura 28 mostra o tempo de resposta médio para cada requisição. Observa-se que, ao aumentar a carga do servidor, o tempo de resposta das requisições também aumenta. Nos gráficos da Figura 29 pode-se observar o impacto do aumento do tempo de resposta no desempenho da reconstrução 3D progressiva. Caixas com mesma cor representam a decodificação do modelo intermediário e os dados de refinamento correspondentes. O espaço em branco entre as caixas da decodificação representa o tempo ocioso no aguardo dos dados de refinamento. A reconstrução do modelo 3D Habacuc requer aproximadamente: 2min no cenário ideal em que há um único acesso por vez; 3,76min quando há 50 acessos simultâneos; e 6,68min quando há 100 acessos simultâneos.

²jmeter.apache.org

³tomcat.apache.org

⁴www.postgresql.org

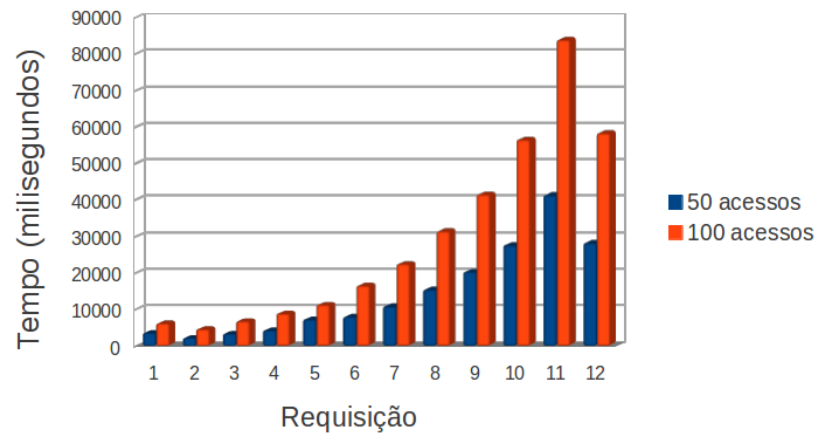
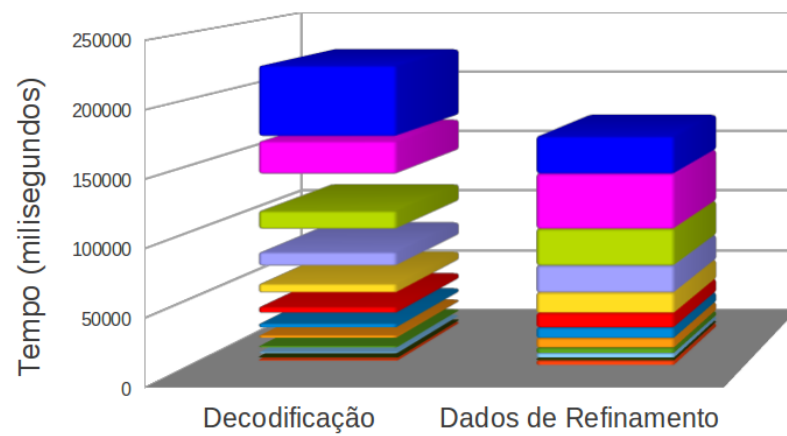
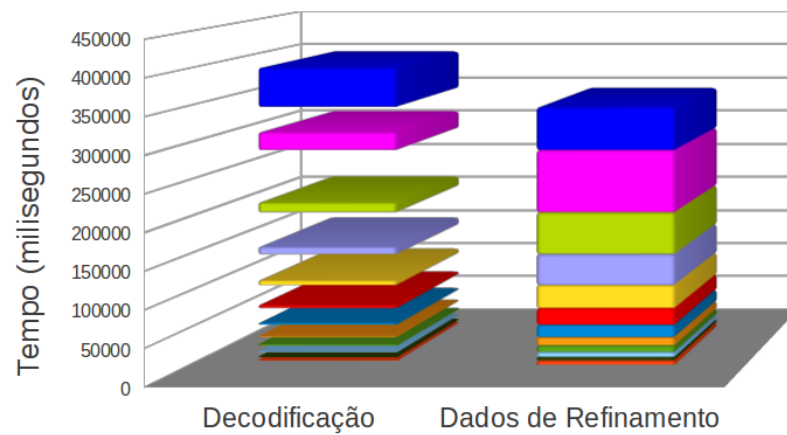


Figura 28 – Tempo de resposta médio de 12 requisições na reconstrução progressiva do modelo 3D Habacuc. Comparação entre 50 e 100 acessos simultâneos ao servidor.



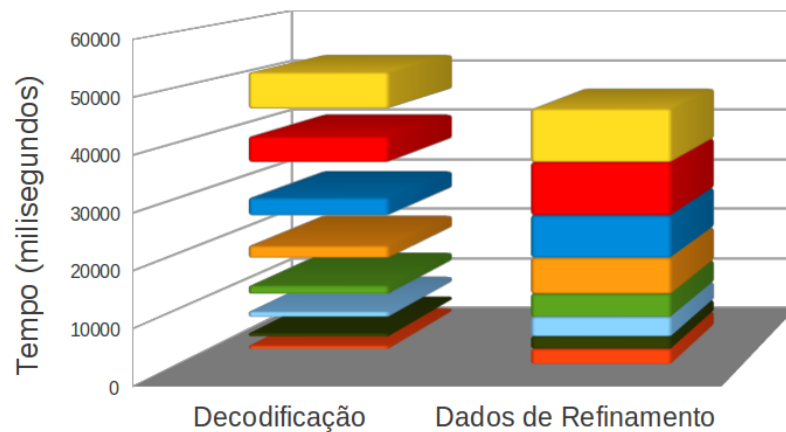
(a)



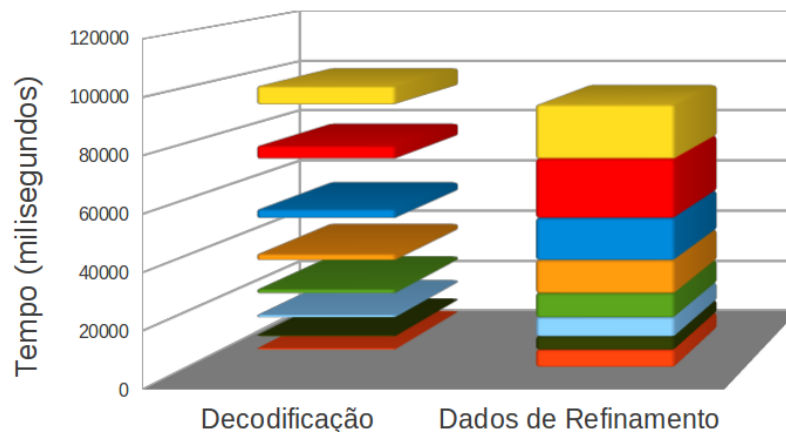
(b)

Figura 29 – Tempo médio da reconstrução total do modelo 3D Habacuc simulando: (a) 50 e (b) 100 acessos simultâneos.

Os resultados visuais dos modelos 3D, como se pode observar nas Figuras 26 e 27, sugerem que os modelos intermediários em torno de 20% de elementos decodificados são visualmente similares ao modelo 3D original. Baseado nessa observação, a reconstrução do modelo 3D Habacuc foi limitada em 20%, equivalente apenas as 8 primeiras requisições. Os gráficos da Figura 30 mostram os tempos de reconstrução médios obtidos: 52,62s quando há 50 acessos; e 1,67min quando há 100 acessos simultâneos. Assim, esta pode ser uma estratégia para manter o bom desempenho da reconstrução 3D progressiva quando há grande quantidade de acessos simultâneos ao servidor. Os benefícios também são significativos no cenário em que há restrições de acesso remoto.



(a)



(b)

Figura 30 – Tempo médio da reconstrução do modelo 3D Habacuc limitada em 20% de elementos simulando: (a) 50 e (b) 100 acessos simultâneos.

6.6 AVALIAÇÃO SUBJETIVA DA QUALIDADE VISUAL PARA A SELEÇÃO DOS NÍVEIS DE RECONSTRUÇÃO

Os experimentos de visualização 3D remota mostraram que o aumento da carga do servidor pode diminuir o desempenho da reconstrução 3D progressiva. Ao disponibilizar modelos 3D realistas, a aplicação *web* precisa considerar os recursos disponíveis no lado do cliente. Se há restrições de memória e processamento, é mais eficiente transmitir o mínimo de dados de geometria e alocar os recursos restantes para os dados de textura. Pan, Cheng e Basu (2005) observaram através de experimentos subjetivos que após a malha alcançar uma certa resolução, aumentar ainda mais a sua resolução não tem impacto visual significativo, enquanto aumentar a resolução da textura continua a melhorar a qualidade visual de todo o objeto. O desafio é definir qual o mínimo de dados de geometria necessários para não haver degradação significativa da qualidade visual.

Em geral, os testes subjetivos são realizados em um ambiente controlado (em laboratório por exemplo) com recursos devidamente alocados (memória, processador, placa de vídeo, etc). Geralmente, um programa *stand-alone* é implementado especialmente para a realização dos testes. Os observadores/participantes são supervisionados por aplicadores, os quais orientam como interagir e responder às perguntas. Por fim, os observadores fornecem sua opinião sobre a qualidade visual do modelo 3D.

Esta seção descreve uma metodologia para a avaliação da qualidade visual subjetiva da reconstrução 3D progressiva como alternativa à avaliação através de métricas objetivas. O objetivo é selecionar os níveis de reconstrução que apresentam qualidade visual suficiente de acordo com a opinião de observadores humanos. As respostas obtidas nos experimentos são armazenadas na base de dados da aplicação e podem ser utilizadas como parâmetro para limitar os níveis de reconstrução em caso de acessos simultâneos ou para facilitar o acesso remoto. As próximas seções descrevem as etapas da metodologia proposta e os resultados obtidos.

6.6.1 Cenário

- **Modelo 3D:** Os modelos 3D são comprimidos progressivamente até obter um modelo base suficientemente pequeno para ser transmitido na Internet.

- **Textura:** Uma textura colorida é aplicada ao modelo 3D utilizando *shaders*. O efeito de mascaramento visual devido ao mapeamento de texturas é desejado, uma vez que um dos objetivos do experimento é determinar qual o mínimo de dados de geometria para prover modelos 3D com boa qualidade visual.
- **Interação:** Os participantes podem interagir livremente com o modelo 3D, através do mouse, *touch screen*, ícones presentes na interface gráfica e teclas de atalho.
- **Área de Exibição:** A área de exibição do modelo 3D ocupa todo o espaço da tela disponível pelo navegador *web* (*full-screen* é permitido).
- **Níveis:** Modelos intermediários a partir do modelo base ao modelo 3D original.
- **Duração:** A duração do experimento inclui o tempo necessário para reconstruir completamente o modelo 3D e depende do tempo do usuário para responder às perguntas.
- **Plano de fundo:** Fundo em preto uniforme.

6.6.2 Procedimento

Uma página *web* apresenta as informações sobre o projeto, objetivos do experimento e a definição de reconstrução 3D progressiva exemplificada através de uma animação. O usuário é orientado a como proceder no experimento: das instruções sobre como interagir com o modelo 3D às perguntas que deverão ser respondidas. Primeiro, o usuário é orientado a realizar uma visualização teste para exercitar a interação 3D. Antes de iniciar o experimento, uma animação do modelo 3D original é apresentada ao usuário, a qual permite visualizar os detalhes do objeto através da ampliação em variados pontos de vista. O participante pode assistir novamente o vídeo dessa animação em qualquer momento durante a realização do experimento sem sair da página de visualização.

A visualização de um modelo 3D é dividida em etapas, em cada qual é solicitada a opinião do participante. Um aviso é exibido na tela informando o início de cada etapa. Tempo após o início da etapa, duas perguntas são feitas ao participante (Figura 31):

1. Você percebeu melhoria na qualidade visual do objeto durante esta etapa?

Percebi pouca melhoria na qualidade.

Percebi melhoria na qualidade.

Percebi grande melhoria na qualidade.

Não percebi diferença na qualidade.

2. Esta qualidade visual está suficiente para você?

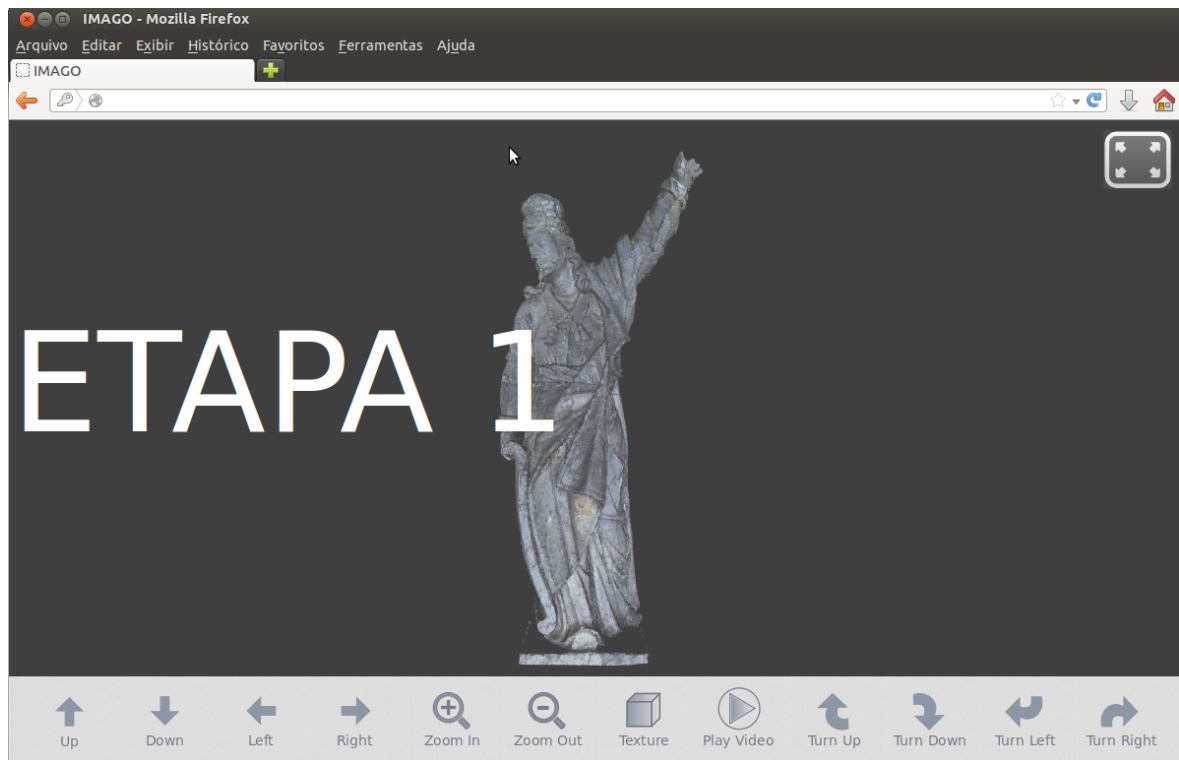
Sim.

Não.

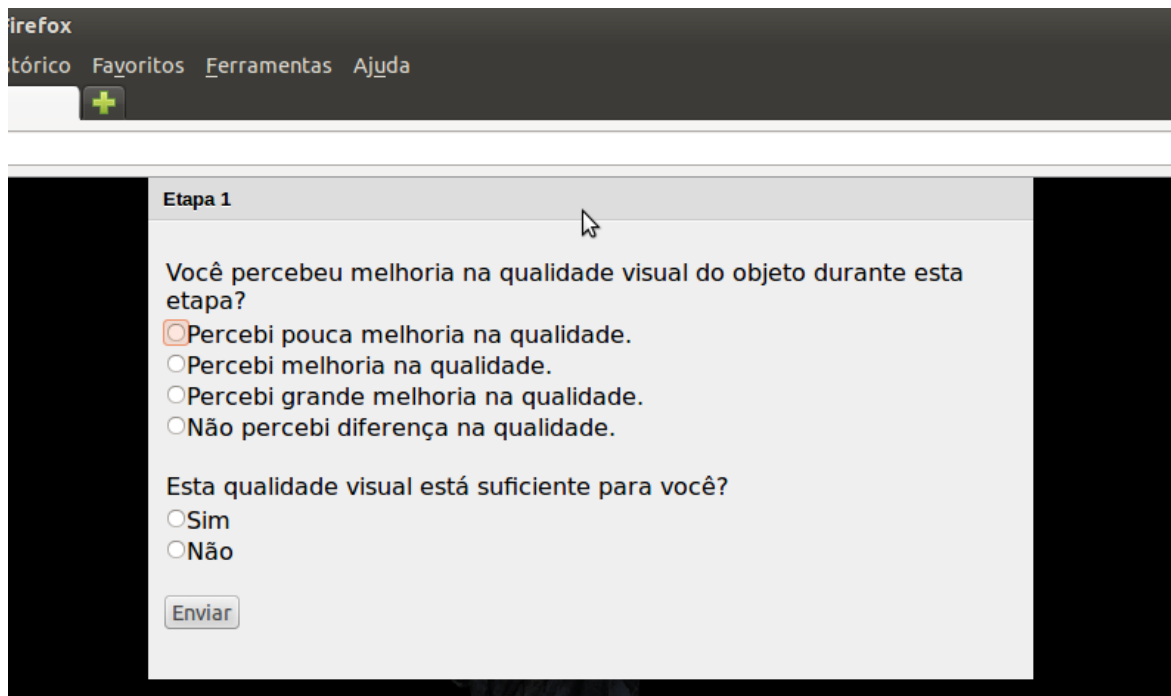
O participante é orientado a marcar a opção “Percebi melhoria na qualidade” caso tenha percebido a melhoria visual, porém não seja capaz de quantificá-la. A primeira pergunta visa capturar a percepção das transições dos modelos intermediários, ou seja, quais transições são facilmente percebidas ou não. Já a segunda, visa avaliar diretamente a satisfação do usuário quanto à qualidade visual do modelo 3D. Para detectar falso positivo nas respostas, uma etapa é adicionada ao experimento na qual não ocorre uma transição de modelos intermediários.

6.6.3 Base de Dados

Os dados obtidos nos experimentos são organizados na base de dados como descrito do diagrama da Figura 32. As informações sobre cada participante são armazenadas na Tabela *tb_observador*. A Tabela *tb_experimento* relaciona o participante e o modelo 3D avaliado. A opinião sobre a qualidade visual dos modelos 3D em cada etapa da visualização é armazenada na Tabela *tb_avaliacao*. Os dados sobre a interação 3D (parâmetros de visualização) também são coletados e armazenados na Tabela *tb_parametro*. Através de consultas ao banco de dados pode-se obter várias informações, *e.g* a partir de qual nível de reconstrução os participantes pouco percebem aumento na qualidade visual ou quais as vistas dos objetos são mais visualizadas.



(a)



(b)

Figura 31 – Realização do experimento para a avaliação da qualidade visual subjetiva da reconstrução progressiva do modelo 3D Habacuc: (a) aviso sobre o início da etapa e (b) perguntas sobre a qualidade visual do modelo 3D.

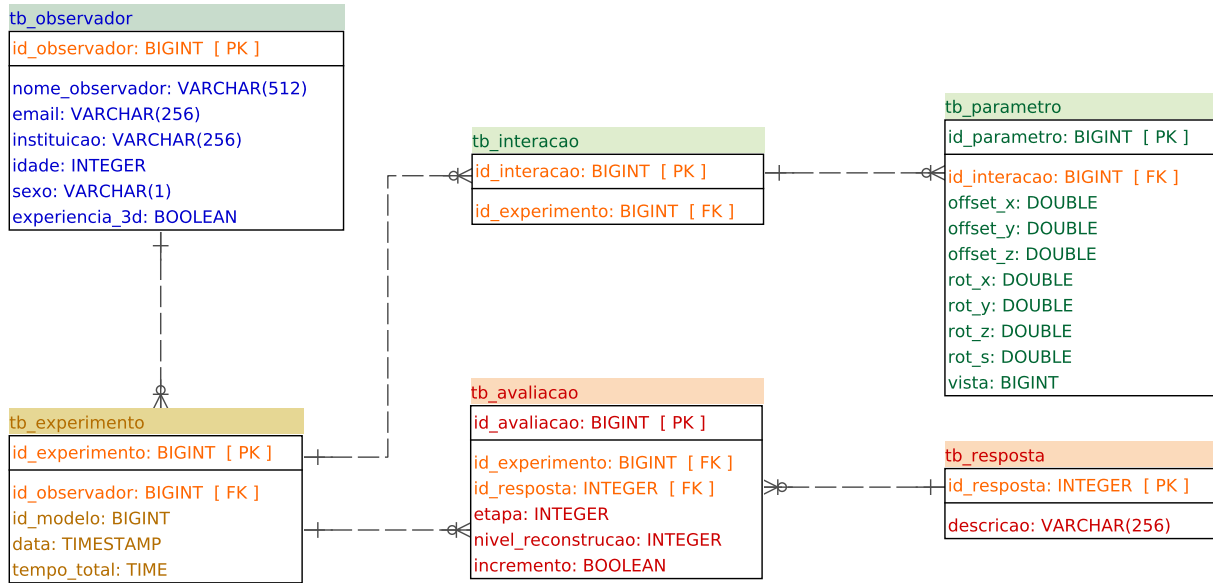


Figura 32 – Diagrama do modelo de dados usado para armazenar informações e dados obtidos no experimento subjetivo.

6.6.4 Seleção das Etapas

Seja $m_0, m_1, m_2, \dots, m_n$ a sequência de $n + 1$ modelos intermediários da reconstrução progressiva, onde m_0 e m_n representam o modelo base e modelo 3D original respectivamente. Uma etapa apresenta a transição visual de $m_i \rightarrow m_j$ tal que $i < j$. A seleção de poucas etapas pode não atender ao objetivo do experimento, enquanto um número grande de etapas pode tornar o teste exaustivo. Transições de modelos intermediários a baixas resoluções com pouca variação de qualidade possuem grandes chances de não serem percebidas pelo usuário, por isso o modelo base não deve apresentar uma degradação excessiva.

6.6.5 Execução

O Algoritmo 3 mostra o pseudo-código da execução do experimento para um modelo 3D com N etapas previamente definidas. Cada etapa assume os seguintes estados: *início*, *pré-transição*, *transição*, *pós-transição* e *coleta*. Ao iniciar uma etapa k , $m_a^{(k)}$ está sendo visualizado e um alerta de início da etapa k é exibido na tela. A execução permanece em estado de *pré-transição* enquanto $m_b^{(k)}$ está sendo requisitado (decodificado). Se $m_b^{(k)}$ está disponível e se $m_a^{(k)}$ foi exibido por pelo menos o tempo mínimo (fixado em 20 segundos), $m_b^{(k)}$ é renderizado (*transição*). $m_b^{(k)}$ então torna-se $m_a^{(k+1)}$ e imediatamente, $m_b^{(k+1)}$ é requisitado em paralelo. Aguarda-se o tempo mínimo

para exibição de $m_b^{(k)}$ (*pós-transição*) para iniciar a coleta dos dados. Enquanto as respostas são aguardadas durante a *coleta*, não é permitida a interação com o modelo 3D. Após o armazenamento das respostas na base de dados, a etapa $k + 1$ inicia.

Algoritmo 3 Pseudo-código do algoritmo para a execução do experimento.

Requer: $TEMPO_MIN$

Requer: $num_etapas \leftarrow N$

```

1:  $k \leftarrow 0$ 
2:  $estado \leftarrow INICIO$ 
3:  $tempo\_transicao \leftarrow 0$ 
4: enquanto  $k < num\_etapas$  faça
5:   se  $estado = INICIO$  então
6:      $k \leftarrow k + 1$ 
7:      $exibe\_modelo(m_a^k)$ 
8:      $exibe\_alerta(k)$ 
9:      $estado \leftarrow PRE\_TRANSICAO$ 
10:     $tempo\_transicao \leftarrow horario\_sistema()$ 
11:  fim se
12:   $tempo\_passou \leftarrow horario\_sistema() - tempo\_transicao$ 
13:  se ( $estado = PRE\_TRANSICAO$ ) e ( $tempo\_passou \geq TEMPO\_MIN$ ) e ( $m_b^k$  esta disponível) então
14:     $estado \leftarrow TRANSICAO$ 
15:     $exibe\_modelo(m_b^k)$ 
16:     $m_a^{k+1} \leftarrow m_b^k$ 
17:     $requisita\_modelo(m_b^{(k+1)})$ 
18:     $tempo\_transicao \leftarrow horario\_sistema()$ 
19:     $estado \leftarrow POS\_TRANSICAO$ 
20:  fim se
21:  se ( $estado = POS\_TRANSICAO$ ) e ( $tempo\_passou \geq TEMPO\_MIN$ ) então
22:     $estado \leftarrow COLETA$ 
23:     $coleta\_dados(k)$ 
24:     $estado \leftarrow INICIO$ 
25:  fim se
26: fim enquanto

```

6.6.6 Resultados

Para a avaliação da qualidade visual subjetiva da reconstrução progressiva do modelo 3D Habacuc foram selecionados quatro participantes com experiência em visualização de gráficos 3D realistas. Todos os 12 modelos intermediários foram submetidos à avaliação, portanto, o experimento contém 13 etapas de visualização (uma etapa a mais para detecção de falso positivo).

A Tabela 18 apresenta os resultados do experimento para cada um dos participantes. O tempo total para a realização do experimento foi de aproximadamente 10min. Os participantes responderam que o objeto apresenta boa qualidade visual (BQV) entre o 6° e o 8° nível de reconstrução, este último corresponde a 20% de elementos decodificados (como observado nos resultados visuais apresentados na Seção 6.5). A opção “Não percebi diferença na qualidade visual” foi a mais escolhida pelos participantes, seguida por “Percebi pouca melhoria na qualidade visual”. Não houve a ocorrência de falso positivo. Baseado neste experimento, conclui-se que a reconstrução progressiva do modelo 3D Habacuc pode ser limitada a partir do 8° nível sem acometer a qualidade visual.

Tabela 18 – Resultado do experimento para avaliação subjetiva da qualidade visual da reconstrução progressiva do modelo 3D Habacuc. BQV é a sigla para “boa qualidade visual”.

Participante	#interações	Tempo	Nível BQV
1	115	9,54min	7
2	190	10,11min	6
3	118	9,23min	7
4	153	9,48min	8

6.7 COMPARAÇÃO COM TRABALHOS SIMILARES

A solução técnica para visualização 3D remota proposta por Lavoué, Chevalier e Dupont (2013) é baseada nos métodos de compressão 3D progressiva de Alliez e Desbrun (2001) e Lee, Lavoué e Dupont (2012). Entre as principais características dessa solução destacam-se a implementação de operações geométricas complexas em JavaScript e a utilização das tecnologias *web* WebGL e HTML5. As principais diferenças deste trabalho em relação ao *framework* proposto são:

- **Contexto de aplicação:** Lavoué *et al.* conduziram seus experimentos a modelos 3D menores, sendo o maior com 540K vértices. O *framework* proposto foi aplicado a modelos 3D realistas obtidos no processo de digitalização 3D, os quais possuem aproximadamente 2-3M vértices. Através dos experimentos pôde-se avaliar o desempenho das recentes tecnologias *web* a condições complexas de memória e processamento.

- **Modelos 3D coloridos:** As vantagens e desvantagens no uso da métrica de cor apresentada em (LEE; LAVOUÉ; DUPONT, 2012) para prevenir a dizimação de vértices que possam causar distorção visual foram discutidas exaustivamente nos Capítulos 4 e 5. O *framework* proposto utiliza a abordagem de compressão 3D progressiva orientada por dados proposta no Capítulo 5.
- **Implementação no lado do servidor:** Na solução proposta por Lavoué *et al.*, os dados de refinamento são armazenados em um arquivo binário P3DW, e não há implementação no lado do servidor para disponibilizar estes dados. A implementação do servidor permite controlar os níveis de reconstrução do modelo 3D. Os limites de desempenho da aplicação *web* do *framework* proposto não foram avaliados em dispositivos móveis. No entanto, o servidor pode identificar esse tipo de acesso e limitar a reconstrução progressiva para viabilizar o acesso remoto por estes dispositivos.

O projeto “Aleijadinho 3D” (RODRIGUES *et al.*, 2013) da Universidade de São Paulo utilizou técnicas avançadas de aquisição e tratamento de malhas para a geração dos modelos 3D das esculturas dos profetas de Aleijadinho. Versões simplificadas dos modelos 3D das esculturas são apresentadas em um cenário virtual representando o adro dos profetas⁵. Destacam-se como contribuição os esforços realizados para a divulgação do patrimônio cultural nacional. Seguem as principais diferenças:

- **Digitalização 3D:** O processo de digitalização 3D realizada pelo projeto Ajeijadinho 3D possui etapas manuais. Por exemplo, o preenchimento de buracos devido à oclusão de regiões onde o *scanner* não alcança foi realizado manualmente com auxílio do MeshLab⁶. O Blender⁷ foi utilizado para modelagem, amostragem, entre outras tarefas. Uma etapa de coloração foi necessária, já que as cores do objeto não foram capturadas na aquisição. Uma empresa especializada em coloração de imagens 3D realizou esta etapa de forma voluntária. Especialistas da empresa inseriram as informações de cor na malha 3D com auxílio de fotos coloridas. Diferentemente, o processo de digitalização 3D realizado pelo grupo IMAGO utilizou um *pipeline* completo que envolve desde a aquisição

⁵www.aleijadinho3d.icmc.usp.br

⁶meshlab.sourceforge.net

⁷www.blender.org

dos dados até a geração de texturas de alta qualidade. Este *pipeline* consiste em uma combinação de algoritmos desenvolvidos por alunos do grupo. A maior parte do processo é automática, e não há informações subjetivas inseridas nos modelos 3D.

- **Preservação Digital 3D:** Apesar dos esforços realizados pelo projeto Aleijadinho 3D, os modelos 3D gerados apresentam restrições quanto a sua utilização para preservação digital devido às características da digitalização 3D empregada. Além da precisão geométrica, as cores originais dos objetos são importantes para as atividades de preservação digital. Informações relevantes sobre os objetos são obtidas através das cores, como por exemplo para a observação de regiões da superfície da escultura com presença de líquens.
- **Visualização dos Modelos 3D:** Os modelos 3D apresentados no cenário real dos adro dos profetas provê uma visão geral da obra de Aleijadinho. Os modelos 3D das esculturas são simplificados na etapa de amostragem para facilitar a disponibilização na *web*. A visualização no navegador *web* é feita através do *plugin* Unity Web Player⁸, disponível para Windows e Mac OS (infelizmente não há suporte oficial para o Linux). O *framework* proposto nesta tese provê a disponibilização de modelos 3D realistas para o estudo remoto dos objetos de forma individual. Além de ser multiplataforma, a aplicação *web* desenvolvida não requer a instalação de *plugins*.

6.8 CONCLUSÃO

Este capítulo apresentou um caso de estudo na visualização *web* de modelos 3D de esculturas feitas pelo artista Aleijadinho. O *framework* de visualização 3D baseada na *web* incorpora a compressão 3D progressiva orientada por dados, uma base de dados de apoio à reconstrução 3D progressiva e uma aplicação *web* para a visualização 3D remota. Os principais benefícios da solução proposta são:

- A visualização interativa de modelos 3D realistas no navegador *web*.
- Não requer a instalação de *plugins*.

⁸unity3d.com

- Representação compacta dos modelos 3D coloridos.
- Transição suave entre os modelos intermediários.
- Baixa latência de visualização.
- Controle dos níveis de reconstrução.
- Possibilita a realização de experimentos subjetivos.

A solução proposta provê a visualização interativa de modelos 3D realistas no navegador *web* sem a necessidade de instalação de *plugins*. Os modelos 3D coloridos são bem compactados na compressão 3D progressiva orientada por dados, a qual provê a transição suave entre os modelos intermediários. A reconstrução 3D progressiva diminui a latência de visualização, uma vez que permite a interação enquanto o modelo 3D é carregado. A estrutura do *framework* permite controlar os níveis de reconstrução para suportar acessos simultâneos ao servidor ou quando há restrições de acesso remoto.

Uma metodologia para a avaliação da qualidade visual subjetiva da reconstrução 3D progressiva foi proposta como alternativa à avaliação através de métricas objetivas. Os dados obtidos nos experimentos podem ser utilizados como parâmetro para a seleção dos níveis de reconstrução. Como trabalho futuro, os experimentos subjetivos podem ser aplicados a diferentes perfis de participantes, incluindo desde estudantes universitários a especialistas em preservação digital. A percepção visual pode variar entre os perfis e assim o controle dos níveis de reconstrução pode ser direcionado ao contexto da visualização: como ferramenta de apoio às atividades de preservação digital ou como uma aplicação voltada à educação e entretenimento (*e.g.* museus virtuais).

7 CONCLUSÕES E TRABALHOS FUTUROS

Esta tese apresentou uma solução efetiva para a visualização de modelos 3D realistas na *web*. A disponibilização desses modelos pode ser dificultada por restrições de acesso remoto, como espaço de armazenamento, memória e capacidade de processamento disponíveis. Ainda que bons computadores e boas conexões de Internet estejam cada vez mais acessíveis, as tecnologias para digitalização 3D fornecem dados cada vez mais precisos. Assim, a relação entre oferta e demanda tecnológica poderá persistir.

O realismo dos modelos 3D gerados no processo de digitalização 3D é devido não apenas à precisão geométrica, mas também à preservação da aparência do objeto, como por exemplo as cores. Os resultados experimentais apresentados no Capítulo 4 comprovam que a preservação das cores contribui para a visualização de modelos 3D realistas na reconstrução 3D progressiva. Os modelos 3D podem ser substituídos por versões menos precisas porém visualmente similares. Esses experimentos possibilitaram uma melhor compreensão sobre os desafios que envolvem a avaliação da qualidade visual de modelos 3D coloridos.

Na revisão do estado da arte foi observado que há poucas soluções dedicadas à compressão progressiva de modelos 3D coloridos. No Capítulo 5 foi apresentada uma nova abordagem de compressão 3D progressiva orientada por dados de mapeamento de textura com o objetivo de melhorar a qualidade visual de modelos 3D coloridos. Além de ser apropriada ao contexto da compressão, a abordagem proposta provê modelos intermediários com boa qualidade visual em relação à preservação das cores, principalmente a baixas resoluções.

O *framework* de visualização 3D baseada na *web* provê a visualização interativa dos modelos 3D realistas na *web* com fidelidade visual requerida no estudo remoto dos objetos. Entre as principais características destacam-se a representação compacta dos modelos 3D coloridos, baixa latência e a visualização 3D no navegador *web* sem a instalação de *plugins*. A estrutura do *framework* possibilita a realização de experimentos subjetivos para a avaliação da qualidade visual dos modelos 3D para a seleção dos níveis de reconstrução.

Através dos resultados experimentais apresentados ao longo desta tese pôde-se comprovar que a preservação das cores combinada à reconstrução 3D progressiva é uma estratégia efetiva e apropriada para a visualização interativa de modelos 3D realistas na *web*. A solução proposta contribui para ampliar a disponibilização de modelos 3D de objetos culturais na Internet. Partindo desta base de referência, diversas funcionalidades podem ser incorporadas à aplicação *web* para apoiar as atividades de preservação digital.

Novas métricas para a avaliação da qualidade visual de modelos 3D coloridos correlacionadas à percepção visual humana precisam ser propostas. O desafio consiste em comparar objetivamente dois modelos 3D coloridos, independente se estes são mapeados por uma textura colorida ou se possuem atributo de cor. Métricas baseadas em imagens podem capturar tanto a geometria do objeto quanto o mascaramento visual proporcionado pela textura, porém é necessário definir um conjunto de vistas suficiente para capturar essas informações. Já para avaliações subjetivas, a interação 3D pode ser investigada como fator de percepção da qualidade visual.

Como trabalho futuro, a etapa final deste trabalho consiste na aplicação efetiva do *framework* proposto. Através de parcerias com museus, como por exemplo o Museu de Arqueologia e Etnologia da Universidade Federal do Paraná, objetos maiores podem ser digitalizados e disponibilizados através da aplicação *web* desenvolvida. A realização de experimentos subjetivos direcionados aos profissionais das áreas relacionadas à preservação digital, como arqueólogos, restauradores, entre outros, possibilita um melhor entendimento sobre como a qualidade visual dos modelos 3D é percebida por esse perfil de usuário.

REFERÊNCIAS

- ALLIEZ, P.; DESBRUN, M. Progressive compression for lossless transmission of triangle meshes. In: **Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques**. New York, USA: ACM, 2001. p. 195–202.
- ANDRADE, B. T.; MENDES, C. M.; JR., J. de O. S.; BELLON, O. R. P.; SILVA, L. 3D preserving xviii century barroque masterpiece: Challenges and results on the digital preservation of aleijadinho's sculpture of the prophet joel. **Journal of Cultural Heritage**, v. 13, n. 2, p. 210–214, 2011.
- APAZA-AGÜERO, K.; SILVA, L.; BELLON, O. R. P. Parameterization and appearance preserving on cubic cells for 3D digital preservation of cultural heritage. In: **Proceedings of the 18th IEEE International Conference on Image Processing**. [S.l.]: IEEE, 2011. p. 2593–2596.
- BENSON, D.; DAVIS, J. Octree textures. **ACM Transactions on Graphics**, v. 21, p. 785–790, 2002.
- BERJÓN, D.; MORÁN, F.; MANJUNATHA, S. Objective and subjective evaluation of static 3D mesh compression. **Image Communication**, v. 28, n. 2, p. 181–195, 2013.
- BIAN, Z.; HU, S.-M.; MARTIN, R. R. Evaluation for small visual difference between conforming meshes on strain field. **Journal of Computer Science and Technology**, v. 24, n. 1, p. 65–75, 2009.
- BULBUL, A.; CAPIN, T.; LAVOUÉ, G.; PREDA, M. Assessing visual quality of 3D polygonal models. **Signal Processing Magazine, IEEE**, v. 28, n. 6, p. 80–90, 2011.
- CABRAL, M.; ZUFFO, M.; GHIROTTI, S.; BELLOC, O.; NOMURA, L.; NAGAMURA, M.; ANDRADE, F.; FARIA, R.; FERRAZ, L. An experience using X3D for virtual cultural heritage. In: **Proceedings of the 12th International Conference on 3D Web Technology**. New York, USA: ACM, 2007. p. 161–164.
- CAI, S.; QI, Y.; SHEN, X. 3D data codec and transmission over the internet. In: **Proceedings of the 12th International Conference on 3D Web Technology**. New York, USA: ACM, 2007. p. 53–56.
- CARROZZINO, M.; BROGI, A.; TECCHIA, F.; BERGAMASCO, M. The 3D interactive visit to Piazza dei Miracoli, Italy. In: **Proceedings of the 2005 ACM SIGCHI International Conference on Advances in Computer Entertainment Technology**. New York, USA: ACM, 2005. p. 192–195.
- CHEN, C.-C.; CHUANG, J.-H. Texture adaptation for progressive meshes. **Computer Graphics Forum**, v. 25, n. 3, p. 343–350, 2006.
- CHENG, I.; BOULANGER, P. A 3D perceptual metric using just-noticeable-difference. In: **Proceedings of the Annual Conference of the European Association for Computer Graphics**. [S.l.: s.n.], 2005. p. 97–100.

- CHENG, W.; OOI, W. T. Receiver-driven view-dependent streaming of progressive mesh. In: **Proceedings of the 18th International Workshop on Network and Operating Systems Support for Digital Audio and Video**. New York, USA: ACM, 2008. p. 9–14.
- CIGNONI, P.; ROCCHINI, C.; SCOPIGNO, R. Metro: Measuring error on simplified surfaces. **Computer Graphics Forum**, v. 17, n. 2, p. 167–174, 1998.
- CIGNONI, P.; SCOPIGNO, R. Sampled 3D models for ch applications: an enabling medium or a technological exercise? **Journal on Computing and Cultural Heritage**, v. 1, n. 1, p. 1–23, 2008.
- CIRIO, G.; LAVOUÉ, G.; DUPONT, F. A framework for data-driven progressive mesh compression. In: **Proceedings of the International Conference on Computer Graphics Theory and Applications**. Lisbon, Portugal: INSTICC Press, 2010. p. 5–12.
- COHEN-OR, D.; LEVIN, D.; REMEZ, O. Progressive compression of arbitrary triangular meshes. In: **Proceedings of the 10th IEEE Visualization Conference**. San Francisco, USA: IEEE Computer Society, 1999. p. 67–72.
- CORSINI, M.; GELASCA, E. D.; EBRAHIMI, T.; BARNI, M. Watermarked 3-D mesh quality assessment. **IEEE Transactions on Multimedia**, v. 9, n. 2, p. 247–256, 2007.
- CRABLE, C. Venus: A unique fast, effective and safe publishing system for high-resolution 3D digital models. In: **Proceedings of the 4th International Conference on Progress in Cultural Heritage Preservation, EuroMed**. [S.l.]: Springer, 2012. (Lecture Notes in Computer Science, v. 7616), p. 644–653.
- EVANS, A.; ROMEO, M.; BAHREHMAND, A.; AGENJO, J.; BLAT, J. 3D graphics on the web: A survey. **Computers & Graphics**, v. 41, p. 43–61, 2014.
- FAROUK, M.; EL-RIFAI, I.; EL-TAYAR, S.; EL-SHISHINY, H.; HOSNY, M.; EL-RAYES, M.; GOMES, J.; GIORDANO, F.; RUSHMEIER, H.; BERNARDINI, F.; MAGERLEIN, K. Scanning and processing 3D objects for web display. In: **Proceedings of the 4th International Conference on 3D Digital Imaging and Modeling**. Washington, USA: IEEE Computer Society, 2003. p. 310–317.
- GANDOIN, P.-M.; DEVILLERS, O. Progressive lossless compression of arbitrary simplicial complexes. **ACM Transactions on Graphics**, v. 21, n. 3, p. 372–379, 2002.
- GRABNER, M.; WOZELKA, R.; MIRCHANDANI, M.; SCHINDLER, K. Web-based visualization of virtual archaeological sites. In: **Proceedings of the 4th International Conference on Virtual Reality, Archaeology and Intelligent Cultural Heritage**. Aire-la-Ville, Switzerland: Eurographics Association, 2003. p. 111–122.
- GUARNIERI, A.; PIROTTI, F.; VETTORE, A. Cultural heritage interactive 3D models on the web: An approach using open source and free software. **Journal of Cultural Heritage**, v. 11, n. 3, p. 350–353, 2010.
- HOPPE, H. Progressive meshes. In: **Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques**. New York, USA: ACM, 1996. p. 99–108.

ISENBURG, M.; SNOEYINK, J. Compressing texture coordinates with selective linear predictions. In: **Computer Graphics International**. Washington, USA: IEEE Computer Society, 2003. p. 126–133.

JIRI, Z.; PAVEL, S. Cultural heritage presentation in virtual environment: Czech experience. In: **Proceedings of the 14th International Workshop on Database and Expert Systems Applications**. [S.l.]: IEEE, 2003. p. 92–96.

KIM, S.-Y.; YOON, Y.-S.; KIM, S.-M.; LEE, K.-H.; HO, Y.-S. Texture coordinate compression for 3-D mesh models using texture image rearrangement. In: HO, Y.-S.; KIM, H. (Ed.). **Advances in Multimedia Information Processing**. Berlin, Heidelberg: Springer, 2005, (Lecture Notes in Computer Science, v. 3767). p. 687–697.

KOLLER, D.; FRISCHER, B.; HUMPHREYS, G. Research challenges for digital archives of 3D cultural heritage models. **Journal on Computing and Cultural Heritage**, v. 2, n. 3, p. 1–17, 2010. ISSN 1556-4673.

KOLLER, D.; TURITZIN, M.; LEVOY, M.; TARINI, M.; CROCCIA, G.; CIGNONI, P.; SCOPIGNO, R. Protected interactive 3D graphics via remote rendering. **ACM Transactions on Graphics**, v. 23, n. 3, p. 695–703, 2004.

KOUTSOUDIS, A.; ARNAOUTOGLU, F.; CHAMZAS, C. On 3D reconstruction of the old city of xanthi. A minimum budget approach to virtual touring based on photogrammetry. **Journal of Cultural Heritage**, v. 8, n. 1, p. 26–31, 2007.

LACOSTE, J.; BOUBEKEUR, T.; JOBARD, B.; SCHLICK, C. Appearance preserving octree-textures. In: **Proceedings of the 5th International Conference on Computer Graphics and Interactive Techniques in Australia and Southeast Asia**. New York, USA: ACM, 2007. p. 87–93.

LARUE, F.; BENEDETTO, M. D.; DELLEPIANE, M.; SCOPIGNO, R. From the digitization of cultural artifacts to the web publishing of digital 3D collections: an automatic pipeline for knowledge sharing. **Journal of Multimedia**, v. 7, n. 2, p. 132–144, 2012.

LAVOUÉ, G. A multiscale metric for 3D mesh visual quality assessment. **Computer Graphics Forum**, v. 30, n. 5, p. 1427–1437, 2011.

LAVOUÉ, G.; CHEVALIER, L.; DUPONT, F. Streaming compressed 3D data on the web using javascript and webgl. In: **International Conference on 3D Web Technology**. New York, USA: ACM, 2013.

LAVOUÉ, G.; CORSINI, M. A comparison of perceptually-based metrics for objective evaluation of geometry processing. **IEEE Transactions on Multimedia**, v. 12, n. 7, p. 636–649, 2010.

LAVOUÉ, G.; Drelie Gelasca, E.; Dupont, F.; Baskurt, A.; Ebrahimi, T. Perceptually driven 3D distance metrics with application to watermarking. In: **Proceedings of SPIE Applications of Digital Image Processing XXIX**. Bellingham, USA: SPIE Press, 2006. v. 6312.

LEE, C. H.; VARSHNEY, A.; JACOBS, D. W. Mesh saliency. In: **ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH)**. New York, USA: ACM, 2005. p. 659–666.

LEE, H.; LAVOUÉ, G.; DUPONT, F. Adaptive coarse-to-fine quantization for optimizing rate-distortion of progressive mesh compression. In: **Vision, Modeling, and Visualization Workshop (VMV)**. [S.l.: s.n.], 2009. p. 73–81.

LEE, H.; LAVOUÉ, G.; DUPONT, F. New methods for progressive compression of colored 3D mesh. In: **Proceedings of 18th International Conference on Computer Graphics, Visualization and Computer Vision. Communication Papers**. [S.l.: s.n.], 2010. p. 199–206.

LEE, H.; LAVOUÉ, G.; DUPONT, F. Rate-distortion optimization for progressive compression of 3D mesh with color attributes. **The Visual Computer**, v. 28, n. 2, p. 137–153, 2012.

LEFEBVRE, S.; DACHSBACHER, C. Tiletrees. In: **Proceedings of ACM Symposium on Interactive 3D Graphics and Games**. New York, USA: ACM, 2007. p. 25–31.

LEFEBVRE, S.; HORNUS, S.; NEYRET, F. Octree textures on the GPU. In: **GPU Gems 2: Programming Techniques for High-Performance Graphics and General-Purpose Computation**. Boston, USA: Addison Wesley, 2005. p. 595–613.

LEVOY, M.; PULLI, K.; CURLESS, B.; RUSINKIEWICZ, S.; KOLLER, D.; PEREIRA, L.; GINZTON, M.; ANDERSON, S.; DAVIS, J.; GINSBERG, J.; SHADE, J.; FULK, D. The digital michelangelo project: 3D scanning of large statues. In: **Proceedings of the 27th Annual Conference on Computer Graphics**. New York, USA: ACM, 2000. p. 131–144.

LINDSTROM, P.; TURK, G. Image-driven simplification. **ACM Transactions on Graphics**, v. 19, n. 3, p. 204–241, 2000.

MAGLO, A.; LEE, H.; LAVOUÉ, G.; MOUTON, C.; HUDELOT, C.; DUPONT, F. Remote scientific visualization of progressive 3D meshes with X3D. In: **Proceedings of the 15th International Conference on Web 3D Technology**. New York, USA: ACM, 2010. p. 109–116.

MENDES, C. M.; SILVA, L.; BELLON, O. R. P. Imago visualization system: An interactive web-based 3D visualization system for cultural heritage applications. **Journal of Multimedia**, v. 7, n. 2, p. 205–210, 2012.

OHBUCHI, R.; MASUDA, H.; AONO, M. Watermarking three-dimensional polygonal models through geometric and topological modifications. **IEEE Journal on Selected Areas in Communications**, v. 16, n. 4, p. 551–560, 1998.

PAJAROLA, R.; ROSSIGNAC, J. Compressed progressive meshes. **IEEE Transactions on Visualization and Computer Graphics**, v. 6, n. 1, p. 79–93, 2000.

PAN, Y.; CHENG, I.; BASU, A. Quality metric for approximating subjective evaluation of 3-D objects. **IEEE Transactions on Multimedia**, v. 7, n. 2, p. 269–279, 2005.

PENG, J.; KUO, C.-C. J. Geometry-guided progressive lossless 3D mesh coding with octree (OT) decomposition. **ACM Transactions on Graphics**, v. 24, n. 3, p. 609–616, 2005.

RAMANARAYANAN, G.; FERWERDA, J.; WALTER, B.; BALA, K. Visual equivalence: Towards a new standard for image fidelity. **ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH)**, v. 26, n. 3, 2007.

ROBLES-ORTEGA, M. D.; FEITO, F. R.; JIMÉNEZ, J. J.; SEGURA, R. J. Web technologies applied to virtual heritage: An example of an iberian art museum. **Journal of Cultural Heritage**, v. 13, n. 3, p. 326–331, 2012.

RODRIGUES, J.; GAZZIRO, M.; GONCALVES, N.; NETO, O.; FERNANDES, Y.; GIMENES, A.; ALEGRE, C.; ASSIS, R. Aleijadinho 3D: tecnologia na difusao e preservacao do patrimonio cultural. In: **Anais do VII SIMP: Convencao do Patrimonio Imaterial: 10 anos depois [2003-2013]**. [S.l.]: Universidade Federal de Pelotas, 2013. p. 1–10.

ROGOWITZ, B. E.; RUSHMEIER, H. E. Are image quality metrics adequate to evaluate the quality of geometric objects? In: **Proceedings of the SPIE, Human Vision and Electronic Imaging**. Bellingham, USA: SPIE Press, 2001. v. 4299, p. 340–348.

ROY, M.; FOUFOU, S.; TRUCHETET, F. Mesh comparison using attribute deviation metric. **International Journal of Image and Graphics**, v. 4, p. 1–14, 2004.

RUSHMEIER, H.; ROGOWITZ, B. E.; PIATKO, C. Perceptual issues in substituting texture for geometry. In: **Proceedings of the SPIE Conference on Human Vision and Electronic Imaging V**. Bellingham, USA: SPIE Press, 2000. v. 3959, p. 372–383.

SANDER, P. V.; SNYDER, J.; GORTLER, S. J.; HOPPE, H. Texture mapping progressive meshes. In: **Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques**. New York, USA: ACM, 2001. p. 409–416.

SCHWARTZ, C.; RUITERS, R.; WEINMANN, M.; KLEIN, R. WebGL-based streaming and presentation of objects with bidirectional texture functions. **Journal on Computing and Cultural Heritage**, v. 6, n. 3, p. 11:1–11:21, 2013.

SILVA, S.; SANTOS, B. S.; MADEIRA, J.; FERREIRA, C. Perceived quality assessment of polygonal meshes using observer studies: A new extended protocol. In: **Proceedings of the SPIE Conference on Human Vision and Electronic Imaging XIII**. Bellingham, USA: SPIE Press, 2008. v. 6806.

TAUBIN, G.; GUÉZIEC, A.; HORN, W.; LAZARUS, F. Progressive forest split compression. In: **Proceedings of the 25th annual conference on Computer graphics and interactive techniques**. New York, USA: ACM, 1998. p. 123–132.

TIAN, D.; ALREGIB, G. Fqm: A fast quality measure for efficient transmission of textured 3D models. In: **Proceedings of the 12th Annual ACM International Conference on Multimedia**. New York, USA: ACM, 2004. p. 684–691.

VALETTE, S.; CHAINE, R.; PROST, R. Progressive lossless mesh compression via incremental parametric refinement. In: **Proceedings of the Symposium on Geometry Processing**. Aire-la-Ville, Switzerland: Eurographics Association, 2009. p. 1301–1310.

WANG, K.; TORKHANI, F.; MONTANVERT, A. Technical section: A fast roughness-based approach to the assessment of 3D mesh visual quality. **Computers & Graphics**, v. 36, n. 7, p. 808–818, 2012.

WATSON, B.; FRIEDMAN, A.; MCGAFFEY, A. Measuring and predicting visual fidelity. In: **Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques**. New York, USA: ACM, 2001. p. 213–220.

WU, J.-H.; HU, S.-M.; TAI, C.-L.; SUN, J.-G. An effective feature preserving mesh simplification scheme based on face constriction. In: **Proceedings of the 9th Pacific Conference on Computer Graphics and Applications**. [S.l.]: IEEE, 2001. p. 12–21.

ZHU, Q.; ZHAO, J.; DU, Z.; ZHANG, Y. Quantitative analysis of discrete 3D geometrical detail levels based on perceptual metric. **Computers & Graphics**, v. 34, n. 1, p. 55–65, 2010.